Identifying and Exploiting Weak-Information Inducing Actions in Solving POMDPs

(Extended Abstract)

Ekhlas Sonu
THINC Lab, Dept. of Computer Science
University of Georgia
Athens, GA. 30602
sonu@cs.uga.edu

Prashant Doshi
THINC Lab, Dept. of Computer Science
University of Georgia
Athens, GA. 30602
pdoshi@cs.uga.edu

ABSTRACT

We present a method for identifying actions that lead to observations which are only weakly informative in the context of partially observable Markov decision processes (POMDP). We call such actions as *weak-* (inclusive of *zero-*) *information inducing*. Policy subtrees rooted at these actions may be computed more efficiently. While zero-information inducing actions may be exploited without error, the quicker backup for weak but non-zero information inducing actions may introduce error. We empirically demonstrate the substantial computational savings that exploiting such actions may bring to exact and approximate solutions of POMDPs while maintaining the solution quality.

Categories and Subject Descriptors

I.2.8 [**Problem Solving, Control Methods, and Search**]: Dynamic Programming

General Terms

Theory, Performance

Keywords

decision making, partial observability, approximation

1. INTRODUCTION

A large body of approximation techniques exploit structure in the problem in order to scale POMDPs [1, 3, 5] leading to significant performance gains for particular problems which exhibit the relevant structure. Consistent with this promising line of investigation, we identify a type of action often found in problem domains such that related computations may be performed more efficiently. Specifically, we consider actions that lead to observations that tend to be only weakly informative. As an example, observations made during movement by a robotic vehicle (typically modeled sequentially post action in a POMDP) tend to be far less informative than those resulting from an action dedicated to observing. We call such actions as *weak-information inducing*; these include those that induce no information as well. We provide a simple and novel definition for weak information-inducing actions, characterizing the weakness of the observations using a parameter. Observing that policy trees rooted at zero-information inducing actions may be compressed, we utilize a simplified backup process that excludes considering observations for any weak-information inducing action while solving POMDPs. This results in significant computational savings, albeit we are currently unable to upper bound the error in optimality that this approximation introduces in the POMDP solution. We demonstrate the significant computational savings by exploiting such actions in the context of an exact solution technique – incremental pruning (IP) [2] – and the well-known point-based value iteration (PBVI) [4], and empirically show that the solutions are of comparable quality.

Cite as: Identifying and Exploiting Weak-Information Inducing Actions in Solving POMDPs (Extended Abstract), Ekhlas Sonu and Prashant Doshi, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1259-1260.

2. λ-INFORMATION INDUCING ACTIONS

We begin by formalizing a definition of such actions and motivation for distinguishing them. We then show how we may exploit such actions thereby reducing the complexity of the backup.

2.1 Definition

In the classical tiger problem, noises subsequent to opening a door (OL/OR) do not provide any information about the door containing the tiger. We generalize this concept to actions leading to weakly informative observations. We call such actions λ-*information inducing*, and define them as:

DEFINITION 1 (λ-INFORMATION INDUCING ACTION). *An action, $a \in A$, is λ-information inducing if for all observations:*

$$1 \leq \frac{max_{s' \in S}\ O(s', a, o)}{min_{s'' \in S}\ O(s'', a, o)} \leq \lambda \quad \forall o \in \Omega$$

where $\lambda \in \mathbb{R}$. We denote the action using a_λ and the set of all such actions using A_λ. Let $\bar{A}_\lambda = A - A_\lambda$.

In general, low values of λ are representative of actions that generate weak observations while high λ signals rich observation(s), although the actual values are subjective to the problem domain.

2.2 Approximate Solution

We may decompose the POMDP belief update into the prediction step where the agent updates its belief based on the action and the correction step where the belief is corrected using the observation that the agent received. We

observe that for zero-information inducing actions ($\lambda = 1$ in Def. 1) the belief updated by the correction step remains unchanged from the prediction step. Hence, we need not perform the correction step for such actions. We extend this to λ-information inducing actions in general.

Our approach is to shorten the belief update process for λ-information inducing actions by ignoring observations. The abbreviated update leads to a different and quicker backup.

Substituting just the prediction step within the Bellman equation leads to the following backup for all actions, $a_\lambda \in A_\lambda$. Let Γ^{n-1} be the set of horizon $n-1$ alpha vectors.

$$\Gamma^{a_\lambda,*} \overset{\cup}{\leftarrow} \alpha^{a_\lambda,*}(s) = R(s,a_\lambda) + \gamma \sum_{s' \in S} T(s,a_\lambda,s')\alpha(s') \;\; \forall \alpha \in \Gamma^{n-1}$$

$$\Gamma_\lambda = \bigcup_{a_\lambda \in A_\lambda} \Gamma^{a_\lambda,*}$$

The backup process proceeds as in the original procedure for all other actions in $\bar{A}_\lambda$ resulting in the set Γ'. We obtain the final set of vectors for horizon n as:

$$\Gamma^n_\lambda = prune\ (\Gamma_\lambda \bigcup \Gamma')$$

Notice the absence of cross-sum operations for actions in A_λ. Consequently, we generate $|\bar{A}_\lambda||\Gamma^{n-1}|^{|\Omega|} + |A_\lambda||\Gamma^{n-1}|$ intermediate vectors in the worst case, which could be far less than $|A||\Gamma^{n-1}|^{|\Omega|}$ vectors generated in the exact approach, if the set A_λ is not empty. The horizon n value function is obtained as: $V^n_\lambda(b) = \max_{\alpha \in \Gamma^n_\lambda} \alpha \cdot b$

3. EXPERIMENTS

We implemented the approximate solution described in Section 2.2 in the context of both IP and PBVI. We selected well-known benchmark problem domains often used to evaluate POMDP solution techniques. In Table 1, we show results for a variety of problem domains. Our methodology was to solve each problem exactly using IP and approximately using PBVI – often for longer time horizon in the latter case. We noted the maximum expected reward obtained by averaging over 1,000 or more random belief points (shown in column R). We then measured the time taken by the approaches modified to exploit λ-information inducing actions to reach the expected rewards obtained previously (including time taken to identify such actions).

4. DISCUSSION

While parameter, λ, in Def. 1 could be seen as a simple way of focusing on actions that induce observations with limited information content, we are unable to bound the difference between the corrected and predicted beliefs for the action in terms of λ. Consequently, the error introduced by the approximation may not be bounded. However, our empirical results in Table 1 indicate that if λ is relatively low, we obtain solutions of quality comparable to the original techniques. We selected IP for demonstration because it is one of the quickest exact POMDP solution techniques, while PBVI is representative of POMDP approximation techniques that scale. If λ is high to the extent that all actions in a problem domain are identified and exploited, the approach may not result in good quality solutions due to high error. Thus, low values of λ that identify a subset of actions are preferable. Consequently, the approach should not be used for problems where the observation functions are identical for most actions.

Method	$\lvert A_\lambda \rvert$	R	Time (secs)	H	$\lvert\Gamma\rvert$	Speedup%
Tiger *(2s, 3a, 2o)*						
IP	n.a.	9.41	3.83 ± 0.2	226	9	
IP + λ=1	2	9.41	3.4 ± 0.22	226	9	∼12
PBVI	n.a.	8.96	0.16 ± 0.2	30	9	
PBVI + λ=1	2	8.96	0.1 ± 0.01	30	9	∼23
Machine_256 *(256s, 4a, 16o)*						
IP	n.a.	1.62	0.08	10	2	
IP + $\lambda = 1$	2	1.62	0.04 ± 0.01	10	2	∼47
PBVI	n.a.	1.33	290.67 ± 1.39	20	1	
PBVI + $\lambda = 1$	2	1.33	164.94 ± 2.26	20	1	∼43
RockSample 5_5 *(801s, 10a, 2o)*						
IP	n.a.	5.7	103.37 ± 0.52	3	151	
IP + $\lambda = 1$	5	5.7	106.36 ± 2.73	3	151	∼3
PBVI	n.a.	8.18	2653.4 ±93.17	9	169	
PBVI + $\lambda = 1$	5	8.18	1954.2 ±8.85	9	169	∼26
RockSample 5_7 *(3201s, 12a, 2o)*						
IP	n.a.	-14.44	2.09 ± 0.02	2	20	
IP + $\lambda = 1$	5	-14.44	1.61 ± 0.02	2	20	∼23
PBVI	n.a.	6.88	3191.6 ± 73.67	4	58	
PBVI + $\lambda = 1$	5	6.88	2410.2 ± 36.21	4	58	∼24
UAV Reconnaissance *(4096s, 9a, 9o)*						
IP	n.a.	–	–	–	–	–
IP + $\lambda = 1$	5	–	–	–	–	–
PBVI	n.a.	–	–	–	–	
PBVI + $\lambda = 1$	5	-8.28	796.13 ± 1.37	2	207	∼80
Learning c2 *(12s, 8a, 3o)*						
IP	n.a.	0.40	0.72	2	338	
IP + λ=10	6	0.39	0.03	2	27	91
PBVI	n.a.	0.63	127.17 ± 3.57	6	873	
PBVI + λ=10	6	0.63	16.65 ± 0.07	7	201	∼87
Learning c3 *(24s, 12a, 3o)*						
IP	n.a.	0.39	54.22 ± 1.93	2	2680	
IP + λ=10	9	0.38	0.77 ± 0.01	2	54	∼99
PBVI	n.a.	0.78	608.94 ± 10.5	8	880	
PBVI + λ=10	9	0.79	158.35 ± 1.79	10	312	∼74
Learning c4 *(48s, 16a, 3o)*						
IP	n.a.	–	–	–	–	
IP + λ=10	12	–	–	–	–	
PBVI	n.a.	0.78	2025.7 ± 41.8	11	896	
PBVI + λ=10	12	0.79	636.39 ± 10.8	12	338	∼69

Table 1: Significant speed ups are obtained for several problems when λ-information inducing actions are exploited for different λ. '–' indicates that the problem could not be solved for at least horizon 2 within an hour. Times are averages of 5 runs on Intel duo 2.8GHz, 4GB RAM.

Acknowledgment This research is partially supported by an NSF CAREER grant, #IIS-0845036.

5. REFERENCES

[1] C. Boutilier and D. Poole. Computing optimal policies for partially observable decision processes using compact representations. In *AAAI*, pages 1168–1175, 1996.

[2] A. R. Cassandra, M. L. Littman, and N. L. Zhang. Incremental pruning: A simple, fast, exact method for partially observable markov decision processes. In *UAI*, 1997.

[3] K.-E. Kim. Exploiting symmetries in pomdps for point-based algorithms. In *AAAI*, pages 1043–1048, 2008.

[4] J. Pineau, G. Gordon, and S. Thrun. Anytime point-based value iteration for large pomdps. *JAIR*, 27:335–380, 2006.

[5] N. Roy, G. Gordon, and S. Thrun. Finding approximate pomdp solutions through belief compression. *JAIR*, 23:1 – 40, 2005.