Escaping Local Optima in POMDP Planning as Inference (Extended Abstract)

Pascal Poupart
David R. Cheriton School of Computer Science
University of Waterloo, Ontario, Canada
ppoupart@cs.uwaterloo.ca

Tobias Lang and Marc Toussaint
Machine Learning and Robotics Lab
FU Berlin, Germany
{tobias.lang,marc.toussaint}@fu-berlin.de

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Miscellaneous

General Terms

Algorithms

Keywords

Planning, POMDPs, EM

1. INTRODUCTION

Planning as inference recently emerged as a versatile approach to decision-theoretic planning and reinforcement learning for single and multi-agent systems in fully and partially observable domains with discrete and continuous variables. Since planning as inference essentially tackles a non-convex optimization problem when the states are partially observable, there is a need to develop techniques that can robustly escape local optima. We propose two algorithms: the first one adds nodes to the controller according to an increasingly deep forward search, while the second one splits nodes in a greedy fashion to improve reward likelihood.[1]

2. PLANNING AS INFERENCE

Consider a partially observable Markov decision process (POMDP) described by a set $\mathcal{S}$ of states s, a set $\mathcal{A}$ of actions a, a set $\mathcal{O}$ of observations o, a transition distribution $\Pr(s'|s,a) = p_{s'|sa}$, an observation distribution $\Pr(o'|s,a) = p_{o'|sa}$ and a reward function $R(s,a) = r_{sa} \in \Re$. An important class of policies (denoted by π) are those representable by a stochastic finite state controller (FSC), which is a directed acyclic graph such that each node n chooses an action a according to $\Pr(a|n) = \pi_{a|n}$, each edge is labeled with an observation o' that chooses a successor node n' according to $\Pr(n'|n,o') = \pi_{n'|no'}$ and the initial node is chosen according to $\Pr(n) = \pi_n$.

Toussaint et al. [6] recently proposed to formulate the optimization of stochastic controllers as a likelihood maximization problem. The idea is to treat rewards as random variables by normalizing them. Let $\bar{R}$ be a binary variable such that $\Pr(\bar{R}{=}true|s,a) = p_{\bar{r}_{true}|sa} = (r_{sa} - r_{min})/(r_{max} - r_{min})$. Similarly, we treat the decision variables A and N as random variables with conditional distributions corresponding to $\pi_{a|n}$, π_n and $\pi_{n'|no'}$. The POMDP is then converted into a mixture of dynamic Bayesian networks (DBNs) where each DBN is t time steps long with a single reward variable at the end and is weighted by a term proportional to γ^t. Hence, the value of a policy is proportional to $p_{\bar{r}_{true}}$ in this mixture of DBNs. To optimize the policy, it suffices to search for the distributions π_n, $\pi_{n'|no'}$ and $\pi_{a|n}$ that maximize $p_{\bar{r}_{true}}$. This can be done by Expectation Maximization (EM), which repeatedly updates the distributions

$$\pi_n^{i+1} \propto \textstyle\sum_s p_s \pi_n^i \beta_{ns}$$
$$\pi_{a|n}^{i+1} \propto \textstyle\sum_{ss'o'n'} \alpha_{sn} \pi_{a|n}^i [p_{\bar{r}_{true}|sa} + \gamma p_{s'|sa} p_{o'|s'a} \pi_{n'|o'n}^i \beta_{n's'}]$$
$$\pi_{n'|o'n}^{i+1} \propto \textstyle\sum_{ss'a} \alpha_{sn} \pi_{a|n}^i p_{s'|sa} p_{o'|s'a} \pi_{n'|o'n}^i \beta_{n's'}$$

Here, $\alpha = \lim_{t\to\infty} \alpha^t$ and $\beta = \lim_{t\to\infty} \beta^t$ are the forward and backward terms obtained in the limit according to

$$\alpha_{s'n'}^t = b_{s'} \pi_{n'} + \gamma \textstyle\sum_{asno'} \alpha_{sn}^{t-1} \pi_{a|n} p_{s'|sa} p_{o'|as'} \pi_{n'|no'}$$
$$\beta_{sn}^t = \textstyle\sum_{as'n'o'} \pi_{a|n} [p_{\bar{r}_{true}|sa} + \gamma p_{s'|sa} p_{o'|as'} \pi_{n'|no'} \beta_{s'n'}^{t-1}]$$

The reformulation of policy optimization as an inference problem opens the door to a variety of inference techniques, however an important problem remains: policy optimization is inherently non-convex and therefore the DBN mixture reformulation does not get rid of local optima issues.

3. ESCAPING LOCAL OPTIMA

Since global optimality is ensured when optimal action and successor node distributions are used for all reachable beliefs, we can perform a forward search from the initial belief to add new nodes each time suboptimal actions or successor nodes are chosen for some reachable beliefs. Since the search grows exponentially with the planning horizon, we propose to start the search from the "mean" beliefs $b_{s|n} \propto \alpha_{sn}$ associated with each node n to reduce the number of steps necessary before a suboptimal action is detected. Alg. 1 describes an incremental forward search that verifies whether the action and successor node distributions are optimal with respect to the value function of the controller for all beliefs reachable at increasing depths. When a non-optimal action or successor node choice is detected, a new node is created with optimal action and successor node distributions. We also create nodes for each belief traversed on the path since their action and successor node distributions may change too. These new nodes are added to the controller, which is re-optimized by EM.

[1] More details can be found in a longer version of this paper.

Cite as: Escaping Local Optima in POMDP Planning as Inference (Extended Abstract), P. Poupart, T. Lang and M. Toussaint, *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. 1263-1264.

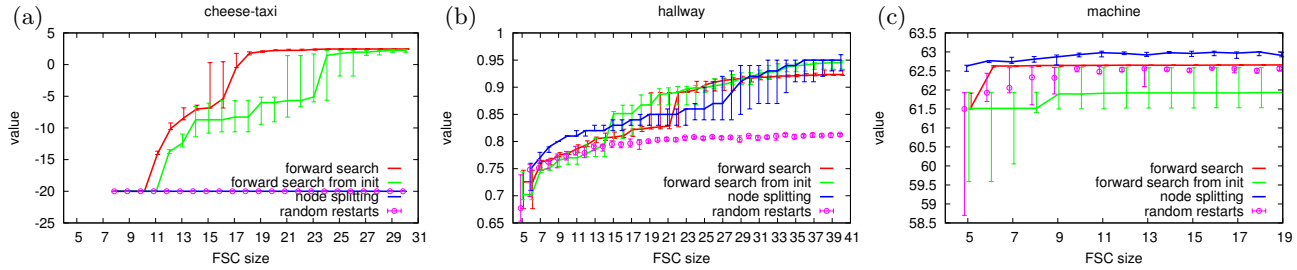

Figure 1: Performance as the number of nodes increases.

Algorithm 1 ForwardSearch(α, β, π, b_0)

for *depth* $= 1$ to ∞ **do**
  for each b reachable from b_0 in *depth* steps **do**
    $v \leftarrow \max_n \sum_s b_s \beta_{sn}$
    $v^* \leftarrow \max_a p_{\tilde{r}_{true}|ba} + \gamma \sum_{o'} p_{o'|ba} \max_{n'} \sum_{s'} b_{s'}^{ao'} \beta_{s'n'}$
    if $v^* - v > 0$ **then**
      return controller with new nodes corresponding to the actions and successor nodes chosen along the path
    end if
  end for
end for

Algorithm 2 NodeSplitting(α,β,π)

for $n \in N$ **do**
  split n into n_1 and n_2
  initialize $\pi_{a|n_1} = \pi_{a|n_2} = \pi_{a|n}$, $\pi_{n'|o'n_1} = \pi_{n'|o'n_2} = \pi_{n'|o'n}$, $\pi_{n_1} + \pi_{n_2} = \pi_n$, $\pi_{n'_1|o'n} + \pi_{n'_2|o'n} = \pi_{n'|o'n}$
  initialize $\alpha_{n_1} + \alpha_{n_2} = \alpha_n$, $\beta_{n_1} = \beta_{n_2} = \beta_n$
  re-run EM
  $gain(n)$ = increase in value when splitting n
end for
return π^*, α^*, β^* based on splitting $n^* = \text{argmax}_n\, gain(n)$

Siddiqi et al. [4] recently proposed an approach to discover the number of hidden states in HMMs by state splitting. In Alg. 2, we adapt this approach to POMDP controllers where internal nodes are split to escape local optima. For each node n of the controller, consider the possibility of splitting that node in two new nodes n_1 and n_2. More precisely replace the parameters that involve n by new parameters that involve n_1 and n_2 and re-run EM. To speed up computation, initialize α and β with those of the unsplitted controller. After re-training the model for each potential split, select the split that yields the largest increase in likelihood.

4. EXPERIMENTS

We tested four methods to escape local optima: i) *forward search* from the mean belief $b_{s|n}$ associated with each node n, ii) *forward search from init* (initial belief), iii) *node splitting* and iv) *random restarts*: retain best controller obtained by running EM from different random initializations. Figure 1 shows the performance of each method as the number of nodes increases for 3 POMDP benchmarks.Each curve is the median of 21 runs from different initial random controllers with error bars corresponding to the 25% and 75% quantiles. The *cheese-taxi* problem is challenging for policy search techniques because its optimal policy includes a long sequence of actions such that any small deviation from that sequence is bad. Only the forward search techniques found good policies because of their ability to modify sequences of actions by adding multiple nodes in one step. For the *hallway* and *machine* problems, adding or splitting one node at a time is adequate, however node splitting outperforms forward search because it evaluates more accurately alternative controllers by re-running EM, which allows it to greedily select the best split at each step.

Table 1: Average value for controllers of different sizes indicated in parenthesis. n.a. = not available.

Techniques	cheese-taxi	hallway	machine
upper bound	2.48	1.19	66.7
HSVI2	2.48	1.03	58.2
biased BPI	2.13 (30)	0.94 (40)	63.0 (30)
QCLP	n.a.	0.72 (08)	61.0 (06)
BBSLS	n.a.	0.80 (10)	n.a.
ForwardSearch	2.47 (19)	0.92 (40)	62.6 (19)
NodeSplitting	-20.0 (30)	0.95 (40)	63.0 (16)

In Table 1, we compare the forward search and node splitting techniques to a leading point-based value iteration technique (HSVI2 [5]) and three policy search techniques for finite state controllers (biased BPI with escape [3], non-linear optimization (QCLP) [1] and stochastic local search (BBSLS) [2]). Since the optimal policy is not known for several problems, we also report an upper bound on the optimal value (computed by HSVI2). The results show that EM with forward search or node splitting is competitive with other policy search techniques. HSVI2 finds better policies, but at the cost of a much larger representation.

5. CONCLUSION

Although there already exists escape techniques for finite state controllers, none of them can be combined with EM (or planning as inference). Hence, this work resolves an important issue by mitigating the effect of local optima and improving the reliability of EM. Our next step is to extend our implementation to factored domains since this is where planning as inference becomes really attractive.

6. REFERENCES

[1] C. Amato, D. Bernstein, and S. Zilberstein. Solving POMDPs using quadratically constrained linear programs. In *IJCAI*, pages 2418–2424, 2007.
[2] D. Braziunas and C. Boutilier. Stochastic local search for POMDP controllers. In *AAAI*, pages 690–696, 2004.
[3] P. Poupart. *Exploiting Structure to efficiently solve large scale partially observable Markov decision processes*. PhD thesis, University of Toronto, 2005.
[4] S. Siddiqi, G. Gordon, and A. Moore. Fast state discovery for HMM model selection and learning. In *AI-STATS*, 2007.
[5] T. Smith and R. Simmons. Point-based POMDP algorithms: improved analysis and implementation. In *UAI*, 2005.
[6] M. Toussaint, S. Harmeling, and A. Storkey. Probabilistic inference for solving (PO)MDPs. Technical Report EDI-INF-RR-0934, School of Informatics, University of Edinburgh, 2006.