Improving Value Function Approximation in Factored POMDPs by Exploiting Model Structure

(Extended Abstract)

Tiago S. Veiga
Institute for Systems Robotics
Instituto Superior Técnico
Universidade de Lisboa
Lisbon, Portugal
tsveiga@isr.ist.utl.pt

Matthijs T. J. Spaan
Delft University of Technology
Delft, The Netherlands
m.t.j.spaan@tudelft.nl

Pedro U. Lima
Institute for Systems Robotics
Instituto Superior Técnico
Universidade de Lisboa
Lisbon, Portugal
pal@isr.ist.utl.pt

ABSTRACT

Linear value function approximation in Markov decision processes (MDPs) has been studied extensively, but there are several challenges when applying such techniques to partially observable MDPs (POMDPs). Furthermore, the system designer often has to choose a set of basis functions. We propose an automatic method to derive a suitable set of basis functions by exploiting the structure of factored models. We experimentally show that our approximation can reduce the solution size by several orders of magnitude in large problems.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence - *Intelligent agents*

General Terms

Algorithms

Keywords

POMDP, Value Function Approximation

1. INTRODUCTION

Partially observable Markov decision processes provide a powerful framework for agent planning under uncertainty [3]. To tackle large problems with many state features, factored POMDP models have become popular, but their solutions can suffer from large value functions. Linear value function approximation has been popular in the literature on fully observable MDPs as a way to compute compact value functions [2]. Extensions to POMDPs have been proposed [1, 7], but are under-explored. In this paper, we propose to exploit the dynamics of the model to generate a suitable set of basis functions for point-based POMDP algorithms. We show in two benchmark problems that our method works in practice and helps to scale up POMDP solving.

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.*

2. LINEAR VALUE FUNCTIONS

A POMDP value function is composed of so-called α-vectors and is piecewise linear and convex [3]. We approximate each α-vector by a linear combination of basis functions. We define a set of allowable vectors $\mathcal{A} \in \mathcal{H} \subseteq \mathbb{R}^{|S|}$ via a set of n_h basis functions $H = \{h_1, \ldots, h_{n_h}\}$. $\mathcal{H}$ is the linear subspace of $\mathbb{R}^{|S|}$ spanned by the set of basis functions H. In factored models each state is represented by a factored state $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, which assigns a value to each state variable X_i, and each basis function's scope is restricted to a subset of variables $C_i \subseteq X$. Then, α-vectors are approximated by $\widetilde{\alpha}$-vectors, written as a linear combination of basis functions h_i:

$$\widetilde{\alpha}(\mathbf{x}) = \sum_{i=1}^{n_h} \omega_{\alpha,i} h_i(\mathbf{c}_i). \tag{1}$$

Point-based POMDP methods compute the maximizing vector at a belief point b with the operator $\texttt{backup}(b)$, which can be extended to approximated linear value functions:

$$\begin{aligned}
\texttt{backup}(b) &= \operatorname*{argmax}_{\{\widetilde{g}_a^b\}_{a\in A}} b \cdot \widetilde{g}_a^b, \quad \text{where} \\
\widetilde{g}_a^b &= R(s,a) + \gamma \sum_o \operatorname*{argmax}_{\{\widetilde{g}_{ao}^k\}} b \cdot \widetilde{g}_{ao}^k, \quad \text{and} \\
\widetilde{g}_{ao}^k(\mathbf{x}) &= \sum_{i=1}^{n_h} \omega_{k,i} \widetilde{g}_{ao}^i(\mathbf{x}), \text{ with} \\
\widetilde{g}_{ao}^i(\mathbf{x}) &= \sum_{\mathbf{x}'} p(o|\mathbf{x}',a) p(\mathbf{x}'|\mathbf{x},a) h_i(\mathbf{c}_i'). \qquad (2)
\end{aligned}$$

We consider an optimized formulation of the backup operator [6], replacing g_{ao} vectors with their approximated version $\widetilde{g}_{ao}$. Inner products may also be performed compactly [7]. Finally, at the end of each backup operation we project the resulting vector back into the space spanned by the basis functions.

3. BASIS FUNCTION CONSTRUCTION

The choice of basis functions is crucial to the success of linear value function approximation. We propose to take advantage of factored models to automatically search for a good set of basis functions.

Note that the scope of each $\widetilde{g}_{ao}^i$ vector is dependent on the scope of basis function h_i and the observation's scope. We define two sets X_o and $X_{\bar{o}}$, which define, respectively,

Algorithm 1 Automatic construction of basis functions

$H \leftarrow nil$
for all $a \in A$ **do**
  $X'_o \leftarrow \Gamma(O_a)$
  $USV^T \leftarrow svd(T_{\mathbf{x}'_o})$
  $H \leftarrow H \cup V$
  for all $X'_j \notin X'_o$ **do**
    $USV^T \leftarrow svd(T_{\mathbf{x}'_j})$
    $H \leftarrow H \cup V$
  $H \leftarrow H \cup R_a$
return linearly independent columns of H

state factors which are in the observation's scope, and those which are not. Mathematically, if we replace basis function scopes in (2), it can be split in two different cases, taking into account that $K = \sum_{\mathbf{x}'_o} p(o|\mathbf{x}'_o, a)p(\mathbf{x}'_o|\Gamma(\mathbf{x}'_o), a)$ is constant, independent of i, and, for any j, $\sum_{\mathbf{x}'_j} p(\mathbf{x}'_j|\Gamma(\mathbf{x}'_j), a) = 1$:

$$\widetilde{g}^i_{ao}(\mathbf{x}) = \begin{cases} \sum_{\mathbf{x}'_o} p(o|\mathbf{x}'_o, a)p(\mathbf{x}'_o|\Gamma(\mathbf{x}'_o), a)h_i(\mathbf{x}'_o) & \text{if } C'_i = X_o \\ K \sum_{\mathbf{x}'_i} p(\mathbf{x}'_i|\Gamma(\mathbf{x}'_i), a)h_i(\mathbf{x}'_i) & \text{otherwise} \end{cases} \quad (3)$$

At this point, we may rewrite the equations in matrix form, where $T_{\mathbf{x}'_j} : p(\mathbf{x}'_j|\Gamma(\mathbf{x}'_j), a)$ and $\Omega_{\mathbf{x}'_j} : p(o|\mathbf{x}'_j, a)$:

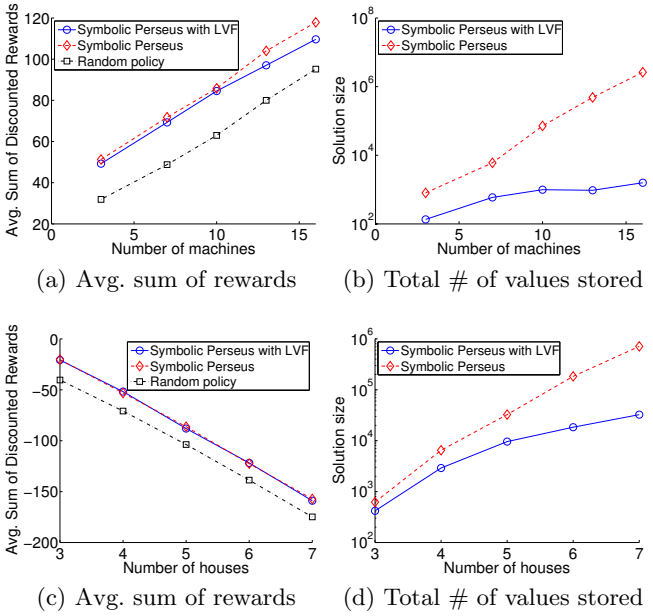
$$\widetilde{\mathbf{g}}^i_{ao} = \begin{cases} \mathbf{T}^a_{\mathbf{x}'_o} \mathbf{\Omega}^{ao}_{\mathbf{x}'_o} \mathbf{h}_i & \text{if } C'_i = X_o \\ K \mathbf{T}^a_{\mathbf{x}'_i} \mathbf{h}_i & \text{otherwise} \end{cases} \quad (4)$$

Both cases represent linear transformations from $\mathbb{R}^{|C'_i|}$ to $\mathbb{R}^{|\Gamma(C'_i)|}$. Geometrically, the result of each of them should be as close as possible to the space spanned by the basis functions, therefore we propose to use the right singular vectors of $\mathbf{T}$ matrices as basis functions. Finally, we include the reward function in H to ensure that rewards can be represented by the set of basis functions. Our methodology is summarized in Algorithm 1.

4. EXPERIMENTS

We implement our ideas in the point-based POMDP solver Symbolic Perseus [5]. We apply our method to the network management problem [5] and to a variation of the fire fighting problem [4], with a fixed number of 2 agents and increasing number of houses. Network management models are run with a belief set of 1000 points and fire fighting models with 500 belief points. Value iteration is run for 50 iterations, and results are averaged over 100 runs of 50 steps each. All sets of basis functions were automatically found by the procedure described in Section 3. We report in Figure 1 the average sum of discounted rewards and solution sizes (computed as the total number of values needed to store the value function) for both domains.

There are gains in solution size up to 3 orders of magnitude in the network domain and more than 1 in the fire fighting domain, which increase with domain sizes. Our method performs better than a random policy and close to plain Symbolic Perseus. There is a small decrease in policy quality, more noticeable in the network domain, but which is expected given the approximate nature of our method.



(a) Avg. sum of rewards (b) Total # of values stored

(c) Avg. sum of rewards (d) Total # of values stored

Figure 1: Network management (a), (b) and fire fighting (c), (d) problems. Comparison between Symbolic Perseus and its extension with linear value function approximation.

5. CONCLUSIONS

Applying linear value function approximation to POMDPs is feasible, although not straightforward. We automatically exploit a problem's structure to derive a good set of basis functions and experimentally test our technique in a point-based method. We show large gains in the solution size for larger problems while maintaining a good policy quality.

Acknowledgments

This work was partially supported by Fundação para a Ciência e Tecnologia (FCT) through grant SFRH/BD/70559/2010 (T.V.), as well as by strategic project UID/EEA/50009/2013.

REFERENCES

[1] C. Guestrin, D. Koller, and R. Parr. Solving factored POMDPs with linear value functions. In *Proceedings IJCAI-01 Workshop on Planning under Uncertainty and Incomplete Information*, 2001.

[2] C. Guestrin, D. Koller, R. Parr, and S. Venkataraman. Efficient solution algorithms for factored MDPs. *JAIR*, 19(1):399–468, Oct. 2003.

[3] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1998.

[4] F. A. Oliehoek, M. T. J. Spaan, S. Whiteson, and N. Vlassis. Exploiting locality of interaction in factored Dec-POMDPs. In *Proc. of AAMAS*, 2008.

[5] P. Poupart. *Exploiting Structure to Efficiently Solve Large Scale Partially Observable Markov Decision Processes*. PhD thesis, University of Toronto, 2005.

[6] G. Shani, P. Poupart, R. I. Brafman, and S. E. Shimony. Efficient ADD operations for point-based algorithms. In *Proc. of ICAPS*, 2008.

[7] T. S. Veiga, M. T. J. Spaan, and P. U. Lima. Point-based POMDP solving with factored value function approximation. In *Proc. of AAAI*, 2014.