A Survey of Point-Based Value Iteration Algorithms for Solving POMDPs

Andrea Bajcsy

Abstract: A key problem in designing autonomous systems is taking input from the environment and producing actions that allow the system to reach a goal. In most real-world settings, actions have stochastic effects on the environment and the state of the environment is only partially observable. Partially Observable Markov Decision Processes (POMDPs) are an expressive and mathematically concrete framework that allows us to optimize decision problems with partial observability. However, POMDPs are hindered by computational intractability. Thus, good approximation techniques are needed in order to leverage the modeling power of POMDPs with realistic computation. The goal of this survey is two-fold: (1) analyze the foundations of solving POMDPs with value iteration and point-based approximations and, (2) analyze a POMDP-based model of human internal state in the context of human-robot interaction.

Introduction & Motivation

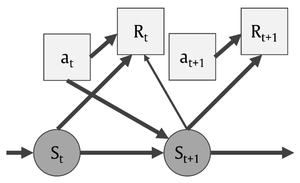
As autonomous systems begin to leave the assembly lines and actually interact with humans, there comes a need to model and predict humans actions and internal belief state in human-robot interaction settings. Partially Observable Markov Decision Processes (POMDPs) provide a solid mathematical framework for optimizing decision problems with partial observability. This is a desired characteristic in robotics applications where autonomous systems often operate under uncertain and dynamic environments.

However, POMDPs are often computationally intractable, taking hours to compute an exact solution even for POMDPs with only a dozen states [3]. This leads to POMDPs being unusable for modeling realistic robotics problems. While computing exact solutions to POMDPs remains difficult, point-based POMDP algorithms have provided fast approximate solutions for POMDPs even with hundreds of states [3].

Background: MDPs & POMDPs

Markov Decision Processes (MDPs)

- S: set of world states
- A: set of actions
- T: state-trans func
- R: reward func

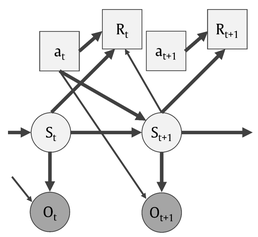


Value Function (discrete):

$$V_t(s) = R(s, \pi_t(s)) + \gamma \sum_{s' \in S} T(s, \pi_t(s), s') V_{t-1}(s')$$

Partially Observable Markov Decision Processes (POMDPs)

- S: set of world states
- A: set of actions
- T: state-trans func
- R: reward func
- O: set of observ.
- Ω: observ. model



Value Function (discrete):

$$V_t(b) = \max_{a \in A} \{ R(b, a) + \gamma \sum_{b' \in B} T(b, a, b') V_{t-1}(b') \}$$

Value Iteration Algorithm

Algorithm 1 Finite-State Value Iteration

1: $V_0(s) = 0, \forall s$
2: **for** $t = 1, 2, ...H$ **do**
3: **for** *all* $s \in S$ **do**
4: $V_{t+1}(s) = \max_a \sum_{s' \in S} T(s, a, s')(V_t(s') + R(s, a, s'))$
5: **end for**
6: **end for**

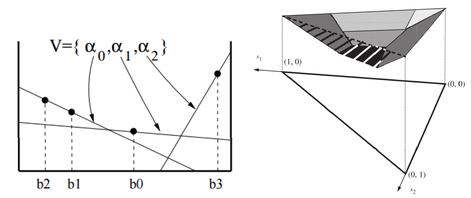
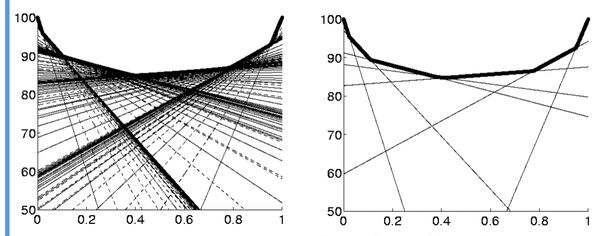


Figure 1: Illustration of 2D and 3D POMDP value functions. Each α-vector forms a hyperplane with dimension equal to the number of states. The optimal value function takes the drawn "bowl" shape. Images from [4] (left), and [2] (right).

Point-Based Value Iteration [4]



Exact value function

Point Based Value Iteration

By keeping alpha-vector for each belief point, PBVI preserves the piece-wise linearity and convexity of the value function. Figures from [6].

Intuition: an agent should only spend time computing solutions to parts of the belief space that can actually be encountered by interacting with the environment

Selecting Belief Points:
- Stochastically simulates step forward for each action to produce new belief set B'
- For each b' in B', measures L1 distance to B
- If b' in B → discards b' point, else keeps b'

Efficient Point-based Value Updates:
- Modifies traditional *Bellman update* such that only one α-vector is kept for each belief point:

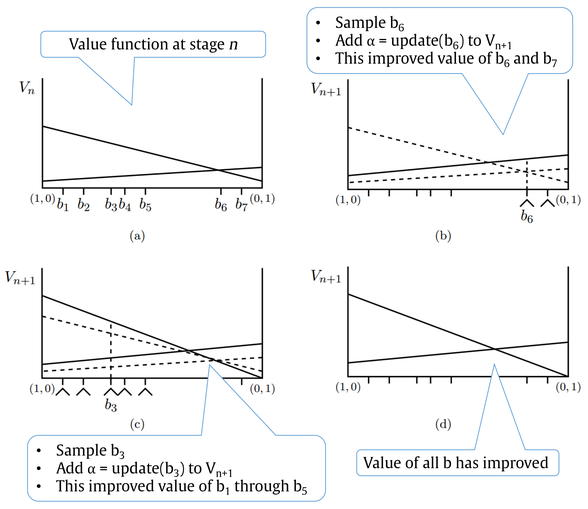
$$V_{t+1}(b) = \max_{a \in A} \{ R(b, a) + \gamma \sum_{b' \in B} T(b, a, b') V_t(b') \}$$

Randomized PBVI for POMDPs [6]

In the PERSEUS algorithm only a random subset of belief points get updated. The claim is that a single update can ultimately improve many points in the belief set.

- **X-axis**: Belief space, **Y-axis:** V(b)
- Tick marks: 7 beliefs in belief space, B
- Solid lines: current α_n^i vectors
- Dashed lines: previous α_{n-1}^i vectors

Compute V_{n+1} from V_n by :



Human-Robot Interaction Systems & PBVI

Conclusion

References

[1] Doshi, F., and Roy, N. Spoken language interaction with model uncertainty: an adaptive human–robot interaction system. *Connection Science 20*, 4 (2008), 299–318.

[2] Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. Planning and acting in partially observable stochastic domains. *Artificial intelligence 101*, 1 (1998), 99–134.

[3] Kurniawati, H., Hsu, D., and Lee, W. S. Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces. In *Robotics: Science and Systems* (2008), vol. 2008, Zurich, Switzerland.

[4] Pineau, J., Gordon, G., Thrun, S., et al. Point-based value iteration: An anytime algorithm for pomdps. In *IJCAI* (2003), vol. 3, pp. 1025–1032.

[5] Porta, J. M., Vlassis, N., Spaan, M. T., and Poupart, P. Point-based value iteration for continuous pomdps. *Journal of Machine Learning Research 7*, Nov (2006), 2329–2367.

[6] Spaan, M. T., and Vlassis, N. Perseus: Randomized point-based value iteration for pomdps. *Journal of artificial intelligence research 24* (2005), 195–220.

[7] Weld, D. Partially observable mdps (pomdps), 2012.