

Planning, Execution & Learning: Planning with POMDPs (II)

Reid Simmons

Solving POMDPs

- Representational Choices
 - Exact V , exact b
 - Witness, policy-iteration, ...
 - Approximate V , exact b
 - Differentiable function approximators (higher-order polynomials, neural nets, ...)
 - Exact V , Approximate b
 - Grid-based
 - Sample-based (PBVI, HSVI)
- **Greedy Approaches** Based on Solving Underlying MDP

Approximating Value Function

- Use Function Approximator with “Better” Properties than Piece-Wise Linear
 - Continuous (differentiable), non-linear
 - Typically use on the order of one vector per action
- Comparisons
 - + Generally much more efficient
 - May poorly represent optimal solution (however, better function approximation usually implies better results)

SPOVA Algorithm (Parr, 1995)

- Approach
 - Use a small set of vectors to represent the value function
 - Approximate the value function by a smooth (differentiable) function

$$V(b) = \max_{v \in \Psi} (v \cdot b)$$

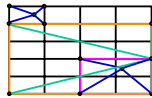
$$\approx \left\{ \sum_{v \in \Psi} v \cdot b \right\}^{1/k}$$
 - Use gradient descent to adjust components of the vectors

$$E(b) = V(b) - \beta \{ \max_{a'} \{ R(a, b) + \gamma \sum_{b'} p(b' | a, b) V(b') \} \}$$

$$v_{i,t+1}(s) = v_{i,t}(s) + \alpha E(b)b(s)(v_i \cdot b)^{k-1} / V(b)^{k-1}$$

Approximating Belief Space

- Use Grid-Based Approximation
 - Discretize belief space: Place finite grid over belief simplex
 - Evaluate value function at grid points
 - Interpolate
- Regular Grid (Lovejoy)
 - + Simple method, easy interpolation
 - Exponential space needed
- Non-Regular Grid (Hauskrecht)
 - + More accurate – tries to follow value contours
 - Interpolation is difficult
- Variable-Resolution Grid (Zhou & Hansen)
 - + Fairly accurate – grid points added where distinctions are needed
 - + Interpolation is fairly easy – add virtual grid points

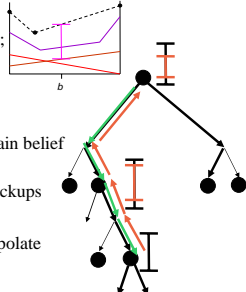


Heuristic-Search Value Iteration (Smith, 2004)

- **Approximate Belief Space**
 - Deals with only a subset of the belief points
 - Focus on the most relevant beliefs
 - Focus on the most relevant actions and observations
- **Main Idea**
 - Value iteration is the dynamic programming form of a tree search
 - Go back to the tree and use heuristics to speed things up
 - But still use the special structure of the value function and plane backups

HSVI Approach

- Maintain Constraints on Value of Beliefs
 - Lower and upper bounds
 - Initialize upper bound to QMDP: Lower bound to "always a"
- Explore the "Horizon" Tree
 - Back up values to further constrain belief values
 - Lower bound is regular plane backups
 - Upper bound is set of points
 - solve linear program to interpolate
 - approximate upper bound



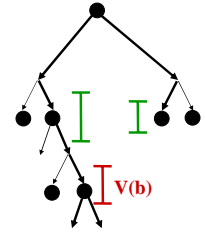
Planning, Execution & Learning: POMDP II

7

Simmons : Spring 2007

HSVI Approach

- Need to decide:
 - When to terminate search?
 - Minimal gain
 - $\text{width}(V(b)) < \epsilon \gamma^t$
 - Which action to choose?
 - Highest upper bound: $\text{argmax}_a Q(b, a)$
 - Which observation to choose?
 - Reduce excess uncertainty most
 - $\text{argmax}_o p(o | b, a) * (\text{width}(V(\tau(b, a, o))) - \epsilon \gamma^{t+1})$



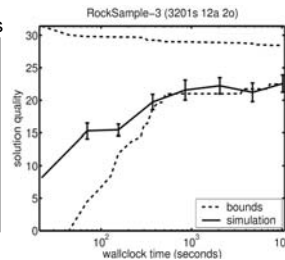
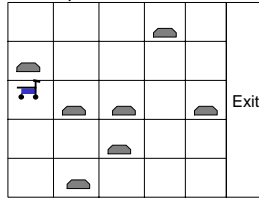
Planning, Execution & Learning: POMDP II

8

Simmons : Spring 2007

HSVI Results

5x5 map, 7 rocks, 3200 states



Planning, Execution & Learning: POMDP II

9

Simmons : Spring 2007

Greedy Approaches to POMDP Planning

- Solve Underlying MDP
 - $\pi_{\text{MDP}}: S \rightarrow A$; $Q_{\text{MDP}}: S \times A \rightarrow \mathfrak{R}$
- Choose Action Based on Current Belief State
 - "most likely" (Nourbakhsh, 1994) - $\pi_{\text{MDP}}(\text{argmax}_s(b(s)))$
 - "voting" (Simmons & Koenig, 1995) - $\text{argmax}_a(\sum_{s \in S} b(s) \delta(a, \pi_{\text{MDP}}(s)))$ where $\delta(a, b) = (1 \text{ if } a=b; 0 \text{ otherwise})$
 - "Q-MDP" (Cassandra, 1995) - $\text{argmax}_a(\sum_{s \in S} b(s) Q_{\text{MDP}}(s, a))$
- Essentially, try to act optimally as if the POMDP were to become observable after the next action
 - **Cannot plan to do actions just to gain information**

Planning, Execution & Learning: POMDP II

10

Simmons : Spring 2007

Greedy Approaches to POMDP Planning

- "Dual-Mode Control" (Cassandra 1996) - Extension to Allow Information-Gathering Actions
 - Compute entropy $H(b)$ of belief state
 - If entropy is below a threshold, use a greedy method $Z(a, b)$ for choosing action
 - If entropy is above a threshold, choose the action that reduces expected entropy the most

$$EE(a, b) = \sum_{b'} p(b' | a, b) H(b')$$

$$\pi(s) = \text{argmax}_a Z(a, b) \quad \text{if } H(b) < t$$

$$\text{argmin}_a EE(a, b) \quad \text{otherwise}$$

Planning, Execution & Learning: POMDP II

11

Simmons : Spring 2007