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

Reid Simmons

Manuela Veloso

Solving POMDPS

- Value Iteration
  - Dynamic programming approach using Bellman equation
- Policy Iteration
  - Represent policy as FSM and incrementally modify
- Point-Based Approaches
  - Solve for a ***single*** initial belief state, rather than all states
  - Iterate until expected value of the given initial belief state converges to within some threshold
  - Amenable to heuristic search
- Greedy Approaches
  - Use solution to underlying MDP
  - Basically assumes that world becomes observable after 1 step

Greedy Approaches

- Solve Underlying MDP
  - $\pi_{MDP}: S \rightarrow A$; $Q_{MDP}: S \times A \rightarrow \Re$
- Choose Action Based on Current Belief State
  - **"most likely"** (Nourbakhsh, 1994) – $\pi_{MDP}(\text{argmax}_s(b(s))$
  - **"voting"** (Simmons & Koenig, 1995) – $\text{argmax}_a(\sum_{s \in S} b(s)\delta(a, \pi_{MDP}(s)))$ where $\delta(a, b) = (1$ if $a=b$; 0 otherwise)
  - **"Q-MDP"** (Cassandra, 1995) – $\text{argmax}_a(\sum_{s \in S} b(s)\, Q_{MDP}(s, a))$
- 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***

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

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

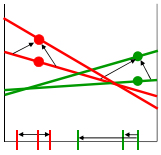
$$\pi(s) = \begin{cases} \text{argmax}_a Z(a, b) & \text{if } H(b) < t \\ \text{argmin}_a EE(a, b) & \text{otherwise} \end{cases}$$

Point-Based Value Iteration (Pineau, 2003)

- ***Approximate Belief Space***
  - Deals with only a subset of the belief points
  - Focus on the most relevant beliefs
  - Focus on the points that add the most information
- Tackles the ***Curse of History***
  - Number of action/observation histories typically grows exponentially
  - Leads to exponential number of vectors in value function

PBVI Approach

1. Maintain One Alpha-Vector Per Belief
2. Perform Backups Over Only Those Vectors
3. Grow Belief Space Intelligently
   - Create only reachable belief points
   - Choose beliefs so as to create uniform density

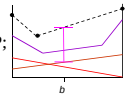


Heuristic-Search Value Iteration (Smith, 2004)

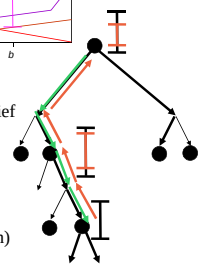
- ***Approximate Belief Space***
  - Deals with only a subset of the belief points
  - Focus on the most relevant beliefs
  - Focus on the most relevant actions and observations
- ***Main Idea***
  - Value iteration is the dynamic programming form of a tree search
  - Go back to the tree and use heuristics 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"

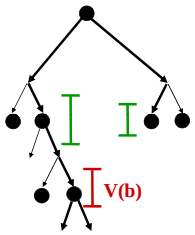


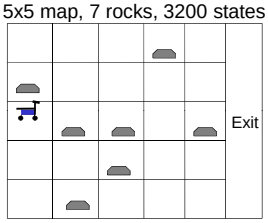
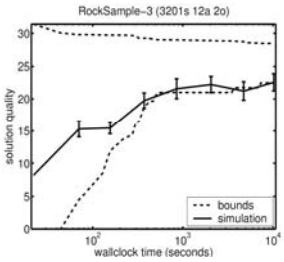
- Explore the "Horizon" Tree
  - Back up values to further constrain belief values
  - Lower bound is regular plane backups
  - Upper bound is set of points
    - solve linear program to interpolate
    - approximate upper bound (sawtooth)



HSVI Approach

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



HSVI Results

5x5 map, 7 rocks, 3200 states