ACRL (16-899C, S09) Lecture #15 (March 12, 2009)

POMDPs Continued and Function Approximation Strategies

Lecturer: Drew Bagnell *Scribe: Andrew Maas*

1 Introduction

We will continue our discussion of POMDPs and some basic strategies for solving them. After reviewing some of the POMDP basics we will move into a more detailed discussion on methods for approximating value funcitons.

2 POMDP Review

- Last time we discussed the basic structure of POMDPs.
- A fundamental strategy is to convert the POMDP to an Information-State(I-State) MDP. There are several simple I-states which are not practical. One we discussed which is practical is the belief I-state. The belief I-state is a distribution over states condintioned on the history.
- We can solve all adaptive control problems optimally! Given our unknown parameters θ, we augment our state to include θ. θ has no dynamics so $\theta_t = \theta_{t+1}$. We then solve the I-state MDP over the belief state which includes θ (our augmented state).
- We can "solve" all POMDPs using various algorithms:
  - Discretize the I-state annd use value iteration. Because the dimension of the I-state is approximately equal to the number of states, this algorithm is exponential in the number of states. It is also $O(T)$, where T is the total time.
  - Policy iteration over trees describing policies. This is $O(exp(T))$ in the worst case, though it might be better in practice.
  - We can solve a special case: Linear Quadratic Gaussian (LQG).
  - Bar-Shalom paper/method is interesting. You should take a look at it.

3 Approximate Strategies: Receeding Horizon Control

Making a policy is too much work. We move now to talking about various methods for side-stepping building a complete policy.

We can use the "poor man's policy" where we re-plan at every timestep to choose our action. Receeding Horizon Control is an extension of this approach where we plan for only a short window, and then execute the planned action before replanning. This approach is used in chess and ZMP preview controllers.

Here are some other strategies we can use which will allow us to avoid building a complete policy:

1. Do certainty equivalence even though it's wrong. This allows for "passive" active learning, where we learn new things because our assumption makes us take incorrect actions.

2. Compute the current belief state and select the action which maximizeds the one-step reward: $\text{argmax}_a \sum_s b(s)r(s,a) + \tilde{v}(b')$ where $\tilde{v}(b') = \sum_{s'} v(s')b'(s')$ and b' is the belief in the subsequent timestep.

   This algorithm only explores if it will help you right now. This doesn't really seek to resolve uncertainty. The resulting value approximation is optimistic about the true value, it is always $\geq v(b)$

3. Another optimistic approach is to assume that all uncertainty will resolve in your favor. This is the approach used by the D* algorithm. At every step will either learn something, or assume the best thing happened. This approach is very good at compressing the belief state (reducing apparent uncertainty)

4. Alternatively we could assume that all uncertainty resolve against us. This produces a pessimistic estimate, in which our estimated values are always $\leq v(b)$

5. Another pessimistic policy is to assume the policy π is fixed at the next timestep. We can then do one step lookahead to choose an action. Thus we choose $\text{argmax}_a \sum_s b(s)r(s,a) + \sum_{s'} b(s')v^\pi(s')$ where v^π is the value function under the policy which we assume is fixed. Bar-Shalom paper is a special case of this approach where the value estimate is quadratic.

   An extension to the above approach is to optimize over π while doing the above. For example, we might take the max over multiple πs, or tweak π for a particular b

6. We can also push back thee horizon (e.g. 2-step lookahead). This is exponential in the number of lookahead steps.

4 Value Function Approximation

Now we want to build up a policy instead of doing receeding horizon control.

- Sum of linear basis functions (future discussion)
- DDP. Lots of dimensions. Very local.
- Grid-based. There are quite a few options such as: adaptive resolution, random grids, etc. Grids in general have dimensionality issues.