L22 – Solving POMDPs Offline

AIMA 17.5, KAlg 20.5-6; 21.4-5

What you should know after this lecture

- How to specify a full POMDP policy
- What an α-vector is
- How to use value iteration to compute a value function
- Point-based value iteration methods find policy concentrated on belief space reachable under optimal policy
- New online belief-space planning methods can be smarter than expectimax and work in complicated domains

Recall POMDP definitions

MDP with added observation process

- MDP has
  - a set of states $\mathcal{S}$
  - a set of actions $\mathcal{A}$
  - transition model
    $T(s, a, s') = P(S_{t+1} = s' \mid S_t = s, A_t = a)$
  - reward function $R : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$
  - discount factor γ (and possibly horizon T)
- POMDP adds
  - a set of observations $\mathcal{O}$
  - observation model
    $O(s', a, o) = P(O_{t+1} = o \mid A_t = a, S_{t+1} = s')$
- optionally
  - an initial distribution over states, $b_0 = P(s_0)$

Recall Bayes filter belief update

Given previous belief b, action a and observation o, what is the new belief $b' = \mathsf{bf}(b, a, o)$?

$$
\begin{aligned}
\mathsf{bf}(b, a, o)(s') &= P(S_{t+1} = s' \mid A_t = a, O_{t+1} = o, B_t = b) \\
&= \frac{1}{\eta} P(O_{t+1} = o \mid S_{t+1} = s', A_t = a, B_t = b) \\
&\quad P(S_{t+1} = s' \mid B_t = b, A_t = a) \\
&= \frac{1}{\eta} O(s', a, o) \sum_s P(S_{t+1} = s' \mid B_t = b, A_t = a, S_t = s) \\
&\quad P(S_t = s \mid B_t = b, A_t = a) \\
&= \frac{1}{\eta} O(s', a, o) \sum_s T(s, a, s') b(s)
\end{aligned}
$$

where $\eta = P(o \mid a, b) = \sum_{\tilde{s}} O(\tilde{s}, a, o) \sum_s T(s, a, \tilde{s}) b(s)$

Value of a policy tree: α vector

Let's start by computing the value of executing a policy tree π in a known starting state s: $V^{\pi}(s)$.

- Base case when π is single node ($H = 1$):

$$V^{\pi}(s) = R(s, \pi.a)$$

- Recursive case: depending on what state s' we transition to and what observation we get (which depends on the state), we will execute one of our subtrees ($\pi(o)$) in s':

$$V^{\pi}(s) = R(s, \pi.a) + \gamma \sum_{s'} T(s, \pi.a, s') \sum_{o} O(s', \pi.a, o) V^{\pi(o)}(s')$$

Let

$$\alpha^{\pi} = \left[V^{\pi}(s^1), \ldots, V^{\pi}(s^{|S|}) \right]$$

Then value at a belief is

$$V^{\pi}(b) = \sum_{s} b(s) V^{\pi}(s) = b \cdot \alpha^{\pi}$$

Infinite-horizon discounted case

- Optimal value function is convex
- It can be piecewise linear or curved! Curve can arise in the limit of infinitely many pieces.
- Value iteration algorithm still works, iteratively computing sets of α vectors.
- Terminate when the change in the maximum difference between subsequent value functions becomes small.
- Cool (advanced topic): if the optimal value function has finitely many pieces, then there is a finite-state machine controller that is optimal!

POMDP "backup"

What is the value, at belief state b, of taking action a and then, for each $o \in \mathcal{O}$, if we get observation o, continuing with a policy whose value is α_o?

$$
\begin{aligned}
V_a(b) &= \left(\sum_s b(s)R(s,a)\right) + \gamma \sum_o P(o \mid b,a)\alpha_o \cdot \mathsf{bf}(b,a,o) \\
&= \left(\sum_s b(s)R(s,a)\right) + \gamma \sum_o P(o \mid b,a) \sum_{s'} (\alpha_o(s')\mathsf{bf}(b,a,o)(s')) \\
&= \left(\sum_s b(s)R(s,a)\right) + \gamma \sum_o P(o \mid b,a) \sum_{s'} \alpha_o(s') \frac{O(s',a,o)\sum_s T(s,a,s')b(s)}{P(o \mid b,a)} \\
&= \sum_s b(s) \left(R(s,a) + \gamma \sum_o \sum_{s'} \alpha_o(s')O(s',a,o)T(s,a,s')\right) \\
&= b \cdot \alpha'
\end{aligned}
$$

We get a new alpha vector! $\textsc{Backup}(a, [\alpha_1, \ldots, \alpha_{|\mathcal{O}|}]) = \alpha'$

POMDP Value Iteration

POMDP-VI$(\mathcal{S}, \mathcal{A}, T, R, \mathcal{O}, O, \gamma)$

1 $\Gamma_0 = \{R(\cdot, a^1), R(\cdot, a^2), \ldots, R(\cdot, a^{|\mathcal{A}|})\}$ // H=1 α vectors
2 $t = 1; \Delta = \infty$
3 **while** $\Delta > \epsilon$ // Stop when max change in value funs is $< \epsilon$
4 $\quad \Gamma_t = \{\}$
5 $\quad$ **for** $a \in \mathcal{A}$ // Try all combinations of subtrees
6 $\quad\quad$ **for** $(\alpha_1, \ldots, \alpha_{|\mathcal{O}|}) \in$ COMBINATIONS$(\Gamma_{t-1}, |\mathcal{O}|)$
7 $\quad\quad\quad \alpha = $ BACKUP$(a, [\alpha_1, \ldots, \alpha_{|\mathcal{O}|}])$
8 $\quad\quad\quad \Gamma_t = \Gamma_t \cup \alpha$
9 $\quad$ // Ideally, prune out dominated α from Γ_t
10 $\quad \Delta = \max_b ((\max_{\alpha_t \in \Gamma_t} \alpha_t \cdot b) - (\max_{\alpha_{t-1} \in \Gamma_{t-1}} \alpha_{t-1} \cdot b))$
11 $\quad t = t + 1$
12 **return** Γ_t

Variations on value iteration

- Guaranteed to converge to optimum but can be very slow because there may be many tiny little “facets” to the value function
- Idea: sample specific points in belief space to control where we spend our computational / approximation effort.

Point-based value iteration

Point-based backup computes new α vector that is guaranteed to be an improvement at b (and probably elsewhere).

PB-BACKUP($POMDP, \Gamma, b$)

1 For $a \in \mathcal{A}$
2 For $o \in \mathcal{O}$
3 $\alpha_{a,o} = \mathsf{argmax}_{\alpha \in \Gamma} \alpha \cdot \mathsf{bf}(b, a, o)$
4 $\alpha_a = \text{BACKUP}(POMDP, a, [\alpha_{a,1}, \ldots, \alpha_{a,|\mathcal{O}|}])$
5 **return** $\mathsf{argmax}_a \alpha_a \cdot b$

Randomly sampling b will converge to optimal V but slow.

PBVI($POMDP$)

1 $\Gamma = \{R(\cdot, a^1), R(\cdot, a^2), \ldots, R(\cdot, a^{|\mathcal{A}|})\}$
2 **while** not tired
3 $b = \text{SAMPLE-B}(\Gamma)$
4 $\Gamma = \Gamma \cup \{\text{PB-BACKUP}(POMDP, \Gamma, b)\}$
5 **return** Γ

SARSOP

Idea: Use b_0 and only try to estimate V well on belief states reachable via optimal policy. V_Γ is a lower bound on V^*.

SARSOP($POMDP, b_0, \epsilon$)

1 $\Gamma = \{R(\cdot, a^1), R(\cdot, a^2), \ldots, R(\cdot, a^{|\mathcal{A}|})\}$
2 Initialize upper bound V_{up} (e.g. with Q_{mdp})
3 Initialize partial expectimax tree T with root b_0
4 **while** $|V_\Gamma(b_0) - V_{up}(b_0)| > \epsilon$
5 $\quad b_1, \ldots, b_k =$ SAMPLE-PATH(Γ, T)
6 $\quad$ For $b_i \in b_1, \ldots b_k$
7 $\quad\quad \Gamma = \Gamma \cup \{$PB-BACKUP$(\Gamma, b_i)\}$
8 $\quad\quad$ Update V_{up} at b_i
9 $\quad\quad$ Update $T(b_i)$
10 $\quad$ PRUNE(Γ)
11 **return** Γ

Hanna Kurniawati, David Hsu, Wee Sun Lee, SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces.

SARSOP sampling: intuition only

To sample a path, start at root of T and generate a path. At a node b

- Select a that maximizes upper bound $Q_{up}(b, a)$
- Select o that maximizes $|V_{up}(\mathsf{bf}(b, a, o)) - V_{\Gamma}(\mathsf{bf}(b, a, o))|$

We try to stay in high-value parts of the state space and to sample in places where our bounds are far apart.
Terminate a sample trajectory if the difference between upper and lower bounds is such that it will have little effect on the gap at b_0.

Modern online solution methods

We looked at an approximate online solution method: most likely observation. Plan under assumption of MLO, replan when we get a different observation.
MLO can be bad when there's a possible very bad outcome of an action that is not highly likely. It will not be "revealed" by the MLO and we will ignore the downside risk.
Can also do expectimax or sparse sampling on the belief MDP.
MCTS offers more focused search:

- POMCP algorithm (Silver et al): labels nodes with o, a histories rather than beliefs — allows approximate belief representations such as particle sets
- DESPOT algorithm (Ye et al): Uses cleverer sampling (some ideas from SARSOP) and variance reduction techniques to be more efficient than POMCP

Next time

- Real RL is a POMDP!
- We'll start with Bandit problems