1 handout: slides
project proposals are due

MDP Wrap-Up

Real-time Dynamic Programming

for a known MDP. which states to update?

initialize U to an upper bound

update U as we follow greedy policy from s_0

$$U(s) \leftarrow R(s) + \gamma \max_a \sum_{s'} T(s,a,s')U(s')$$

states that agent is likely to visit (nice anytime profile)

Summary of MDP Solving

- value iteration: compute U^{π^*}
  - prioritized sweeping
  - RTDP
- policy iteration: compute U^{π} using
  - linear algebra (exact)
  - simplified value iteration (exact and faster?)
  - modified PI (a few updates, so inexact)

Model-based Reinforcement Learning

Adaptive Dynamic Programming

'model-based'. active vs passive

learn T and R as we go, calculating π using MDP methods (eg, VI or PI)

Until *max-update* $\leq loss - bound \frac{(1-\gamma)^2}{2\gamma^2}$

for each state s

$U(s) \leftarrow R(s) + \gamma \max_a \sum_{s'} T(s,a,s')U(s')$

$$\pi(s) = \operatorname*{argmax}_a \sum_{s'} T(s,a,s')U(s')$$

Prioritized Sweeping

given an experience (s, a, s', r),
  update model
  update s
  repeat k times:
    do highest priority update

to update state s with change δ in $U(s)$:
  update $U(s)$
  priority of $s \leftarrow 0$
  for each predecessor s' of s:
    priority $s' \leftarrow$ max of current and $\max_a \delta \hat{T}(s', as')$

repeat until π doesn't change:
  given π, compute $U^{\pi}(s)$ for all states
  given U, calculate policy by one-step look-ahead

If π doesn't change, U doesn't either.
We are at an equilibrium ($=$ optimal π)!

Exploration vs Exploitation

problem:

Exploration vs Exploitation

problem: greedy (local minima)

$$U^+(s) \leftarrow R(s) + \gamma \max_a f\left(\sum_{s'} T(s,a,s')U^+(s'), N(a,s)\right)$$

where $f(u,n) = R_{\max}$ if $n < k$, u otherwise

Break

- asst 4
- final papers: writing-intensive

Model-free Reinforcement Learning

Q-Learning

$$U(s) \quad = \quad R(s) + \gamma \max_a \sum_{s'} T(s,a,s')U(s')$$

Q-Learning

$$
\begin{aligned}
U(s) &= R(s) + \gamma \max_a \sum_{s'} T(s,a,s')U(s') \\
Q(s,a) &= \gamma \sum_{s'} \left(T(s,a,s')(R(s') + \max_{a'} Q(s',a')) \right)
\end{aligned}
$$

Given experience $\langle s, a, s', r \rangle$:

Q-Learning

$$
\begin{aligned}
U(s) &= R(s) + \gamma \max_a \sum_{s'} T(s,a,s')U(s') \\
Q(s,a) &= \gamma \sum_{s'} \left(T(s,a,s')(R(s') + \max_{a'} Q(s',a')) \right)
\end{aligned}
$$

Given experience $\langle s, a, s', r \rangle$:

$$
Q(s,a) \leftarrow Q(s,a) + \alpha(error)
$$

Q-Learning

$$U(s) = R(s) + \gamma \max_a \sum_{s'} T(s,a,s')U(s')$$

$$Q(s,a) = \gamma \sum_{s'} \left(T(s,a,s')(R(s') + \max_{a'} Q(s',a')) \right)$$

Given experience $\langle s, a, s', r \rangle$:

$$Q(s,a) \leftarrow Q(s,a) + \alpha(error)$$
$$Q(s,a) \leftarrow Q(s,a) + \alpha(sensed - predicted)$$

$$U(s) = R(s) + \gamma \max_a \sum_{s'} T(s,a,s') U(s')$$

$$Q(s,a) = \gamma \sum_{s'} \Big(T(s,a,s')(R(s') + \max_{a'} Q(s',a')) \Big)$$

Given experience $\langle s, a, s', r \rangle$:

$$\begin{aligned}
Q(s,a) &\leftarrow Q(s,a) + \alpha(error) \\
Q(s,a) &\leftarrow Q(s,a) + \alpha(sensed - predicted) \\
Q(s,a) &\leftarrow Q(s,a) + \alpha(\gamma(r + \max_{a'} Q(s',a')) - Q(s,a))
\end{aligned}$$

$$
\begin{aligned}
U(s) &= R(s) + \gamma \max_a \sum_{s'} T(s,a,s')U(s') \\
Q(s,a) &= \gamma \sum_{s'} \Big(T(s,a,s')(R(s') + \max_{a'} Q(s',a')) \Big)
\end{aligned}
$$

Given experience $\langle s, a, s', r \rangle$:

$$
\begin{aligned}
Q(s,a) &\leftarrow Q(s,a) + \alpha(error) \\
Q(s,a) &\leftarrow Q(s,a) + \alpha(sensed - predicted) \\
Q(s,a) &\leftarrow Q(s,a) + \alpha(\gamma(r + \max_{a'} Q(s',a')) - Q(s,a))
\end{aligned}
$$

$\alpha \approx 1/N$?

policy: choose random with probability $1/N$?

Summary

Model known (solving MDP):

- value iteration
- policy iteration: compute U^{π} using
  - linear algebra
  - simplified value iteration
  - a few updates (modified PI)

Model unknown (RL):

- ADP using
  - value iteration
  - a few updates (eg, prioritized sweeping)
- Q-learning

EOLQs

- What question didn't you get to ask today?
- What's still confusing?
- What would you like to hear more about?

Please write down your most pressing question about AI and put it in the box on your way out.
Thanks!