6.825 Techniques in Artificial Intelligence

Markov Decision Processes

- Framework
- Markov chains
- MDPs
- Value iteration
- Extensions

MDP Framework

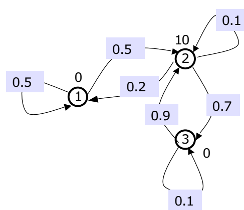
- S : states
- A : actions
- $\Pr(s_{t+1} \mid s_t, a_t)$: transition probabilities
  $= \Pr(s_{t+1} \mid s_0 \ldots s_t, a_0 \ldots a_t)$ Markov property
- R(s) : real-valued reward

Find a policy: $\pi: S \to A$

Maximize
- Myopic: $E[r_t \mid \pi, s_t]$ for all s
- Finite horizon: $E[\sum_{t=0}^{k} r_t \mid \pi, s_0]$
  - Non-stationary policy: depends on time
- Infinite horizon: $E[\sum_{t=0}^{\infty} \gamma^t r_t \mid \pi, s_0]$
  - $0 < \gamma < 1$ is discount factor
  - Optimal policy is stationary

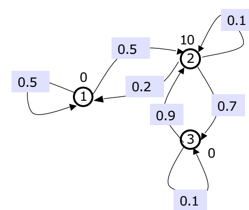
Markov Chain

- Markov Chain
  - states
  - transitions
  - rewards
  - no actions



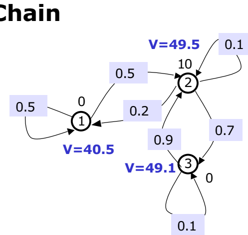
Markov Chain

- Markov Chain
  - states
  - transitions
  - rewards
  - no actions
- Value of a state, using infinite discounted horizon
  $V(s)=R(s)+\gamma\sum_{s'}P(s'|s)V(s')$



Markov Chain

- Markov Chain
  - states
  - transitions
  - rewards
  - no actions
- Value of a state, using infinite discounted horizon
  $V(s)=R(s)+\gamma\sum_{s'}P(s'|s)V(s')$
- Assume $\gamma=0.9$
  V(1)= 0+ .9(.5 V(1)+.5 V(2))
  V(2)=10+ .9(.2 V(1)+.1 V(2)+.7 V(3))
  V(3)= 0+ .9(.9 V(2)+.1 V(3))



Finding the Best Policy

- MDP + Policy = Markov Chain
  - MDP = the way the world works
  - Policy = the way the agent works
- $V^*(s)=R(s) + \max_a[\gamma \sum_{s'} P(s' \mid s, a) V^*(s')]$
- Theorem: There is a unique V^* satisfying these equations
- $\pi^*(s)=\arg\max_a \sum_{s'} P(s' \mid s, a) V^*(s')]$

Computing V^*

- Approaches
  - Value iteration
  - Policy iteration
  - Linear programming

Value Iteration

Initialize $V^0(s)=0$, for all s
Loop for a while [until $\|V^t - V^{t+1}\| < \varepsilon(1-\gamma)/\gamma$]
  Loop for all s
    $V^{t+1}(s) = R(s) + \max_a \gamma \sum_{s'} P(s' \mid s, a)\, V^t(s)$

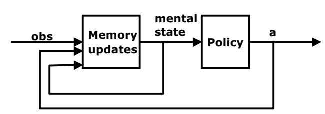
- Converges to V^*
- No need to keep V^t vs V^{t+1}
- Asynchronous (can do random state updates)
- Assume we want $\|V^t - V^*\| = \max_s |V^t(s) - V^*(s)| < \varepsilon$
- Gets to optimal policy in time polynomial in |A|, |S|, $1/(1-\gamma)$

Big state spaces

- Function approximation for V
  - neural nets
  - regression trees
  - factored representations (represent Pr(s'|s,a) using Bayes net)

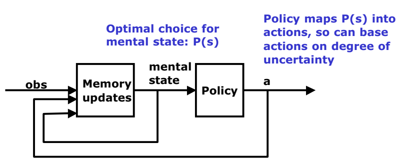
Partial Observability

- MDPs assume complete observability (can always tell what state you're in)
  - POMDP (Partially Observable MDP)
  - Observation: Pr(O|s,a) [O is observation]
  - o, a, o, a, o, a



Partial Observability

- MDPs assume complete observability (can always tell what state you're in)
  - POMDP (Partially Observable MDP)
  - Observation: Pr(O|s,a) [O is observation]
  - o, a, o, a, o, a



Worrying too much

- Assumption that every possible eventuality should be taken into account
- sample-based planning: with short horizon in large state space, planning should be independent of state-space size

Leading to Learning

MDPs and value iteration are an important foundation of reinforcement learning, or learning to behave