Decision Theory: Markov Decision Processes

CPSC 322 Lecture 34

April 11, 2007
Textbook §12.5
Lecture Overview

1 Recap

2 Rewards and Policies

3 Value Iteration
Markov Decision Processes

An MDP is defined by:

- set $S$ of states.
- set $A$ of actions.
- $P(S_{t+1}|S_t, A_t)$ specifies the dynamics.
- $R(S_t, A_t, S_{t+1})$ specifies the reward. The agent gets a reward at each time step (rather than just a final reward).
  - $R(s, a, s')$ is the reward received when the agent is in state $s$, does action $a$ and ends up in state $s'$. 
A Markov decision process augments a stationary Markov chain with actions and values:
Lecture Overview

1 Recap

2 Rewards and Policies

3 Value Iteration
Suppose the agent receives the sequence of rewards $r_1, r_2, r_3, r_4, \ldots$. What value should be assigned?

- **total reward** $V = \sum_{i=1}^{\infty} r_i$
- **average reward** $V = \lim_{n \to \infty} \frac{r_1 + \cdots + r_n}{n}$
- **discounted reward** $V = \sum_{i=1}^{\infty} \gamma^{i-1} r_i$
  - $\gamma$ is the discount factor
  - $0 \leq \gamma \leq 1$
Policies

- A **stationary policy** is a function:

  \[ \pi : S \rightarrow A \]

  Given a state \( s \), \( \pi(s) \) specifies what action the agent who is following \( \pi \) will do.

- An **optimal policy** is one with maximum expected value
  
  - we’ll focus on the case where value is defined as discounted reward.

- For an MDP with stationary dynamics and rewards with infinite or indefinite horizon, there is always an optimal stationary policy in this case.
Value of a Policy

- $Q^\pi(s, a)$, where $a$ is an action and $s$ is a state, is the expected value of doing $a$ in state $s$, then following policy $\pi$.
- $V^\pi(s)$, where $s$ is a state, is the expected value of following policy $\pi$ in state $s$.
- $Q^\pi$ and $V^\pi$ can be defined mutually recursively:

$$V^\pi(s) = Q^\pi(s, \pi(s))$$
$$Q^\pi(s, a) = \sum_{s'} P(s'|a, s) \left( r(s, a, s') + \gamma V^\pi(s') \right)$$
Value of the Optimal Policy

- $Q^*(s, a)$, where $a$ is an action and $s$ is a state, is the expected value of doing $a$ in state $s$, then following the optimal policy.
- $V^*(s)$, where $s$ is a state, is the expected value of following the optimal policy in state $s$.
- $Q^*$ and $V^*$ can be defined mutually recursively:

$$Q^*(s, a) = \sum_{s'} P(s'|a, s) \left( r(s, a, s') + \gamma V^*(s') \right)$$

$$V^*(s) = \max_a Q^*(s, a)$$

$$\pi^*(s) = \arg \max_a Q^*(s, a)$$
Lecture Overview

1. Recap
2. Rewards and Policies
3. Value Iteration
Value Iteration

- **Idea**: Given an estimate of the $k$-step lookahead value function, determine the $k+1$ step lookahead value function.
- Set $V_0$ arbitrarily.
  - e.g., zeros
- Compute $Q_{i+1}$ and $V_{i+1}$ from $V_i$:

  \[
  Q_{i+1}(s, a) = \sum_{s'} P(s'|a, s) \left( r(s, a, s') + \gamma V_i(s') \right)
  \]

  \[
  V_{i+1}(s) = \max_a Q_{i+1}(s, a)
  \]

- If we intersect these equations at $Q_{i+1}$, we get an update equation for $V$:

  \[
  V_{i+1}(s) = \max_a \sum_{s'} P(s'|a, s) \left( r(s, a, s') + \gamma V_i(s') \right)
  \]
Pseudocode for Value Iteration

```
procedure value_iteration( P, r, \theta )
inputs:
P is state transition function specifying \text{P}(s'|a, s)
r is a reward function \text{R}(s, a, s')
\theta a threshold \theta > 0
returns:
\pi [s] approximately optimal policy
V[s] value function
data structures:
V_k[s] a sequence of value functions
begin
  for k = 1 : \infty
    for each state s
      V_k[s] = \max_a \sum_{s'} P(s'|a, s)(\text{R}(s, a, s') + \gamma V_{k-1}[s'])
    if \forall s |V_k(s) - V_{k-1}(s)| < \theta
      for each state s
        \pi (s) = \arg \max_a \sum_{s'} P(s'|a, s)(\text{R}(s, a, s') + \gamma V_{k-1}[s'])
      return \pi, V_k
  end
end
```