Complex Decision Making

CS 470 Introduction To Artificial Intelligence

Daqing Yi

Department of Computer Science
Brigham Young University

Outline

1. Sequential Decision Problems
   - Sequential Markov Process
2. Value Iteration
   - Algorithm
3. Policy Iteration
   - Algorithm

Outline

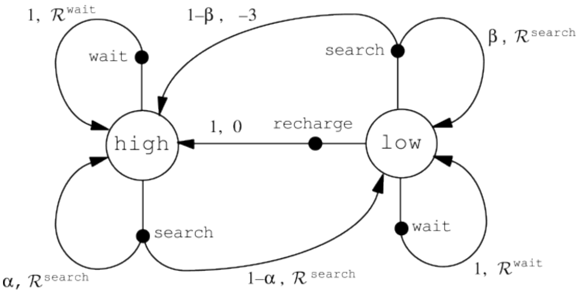
1. Sequential Decision Problems
   - Sequential Markov Process
2. Value Iteration
   - Algorithm
3. Policy Iteration
   - Algorithm

Markov Decision Process

- A set of states S
- A set of actions A
- transition model $P(s' \mid s, a)$
- reward function $R(s)$
- start state s_0

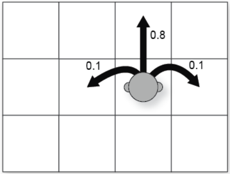
Markov Decision Process

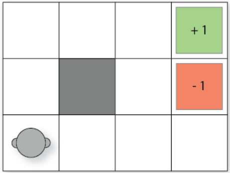
Example - Recycling robot

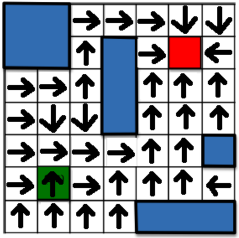


Markov Decision Process

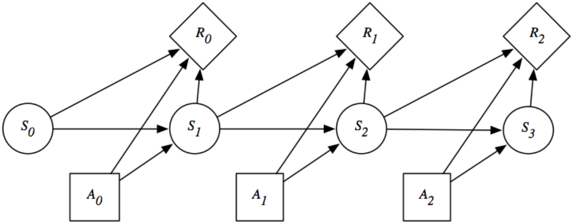
From utility to decision







Utilities over Time



- **additive reward**

$$U_h([s_0, s_1, s_2, \cdots]) = R(s_0) + R(s_1) + R(s_2) + \cdots$$

- **discounted reward**
  - discount factor $\gamma < 1$

$$U_h([s_0, s_1, s_2, \cdots]) = R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots$$

Utilities over Time

- finite horizon
  - if the agent is guaranteed to get to one eventually
- infinite horizon

$$U_h([s_0, s_1, s_2, \cdots]) = \sum_{t=0}^{\infty} \gamma^t R(s_t) \leq \sum_{t=0}^{\infty} \gamma^t R_{max} = \frac{R_{max}}{1-\gamma}$$

Optimal policy

- policy π
- expected utility by executing π starting in s

$$U^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^{t} R(S_t)\right]$$

- optimal policy π_s^* starting in s

$$\pi_s^* = \arg\max_{\pi} U^{\pi}(s)$$

Optimal policy

- long-term reward $U(s)$
- short-term reward $R(s)$
- select actions by maximum expected utility

$$\pi^*(s) = \arg\max_{a \in A(s)} \sum_{s'} P(s' \mid s, a) U(s')$$

Outline

Bellman equation

$$U(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s' \mid s, a) U(s')$$

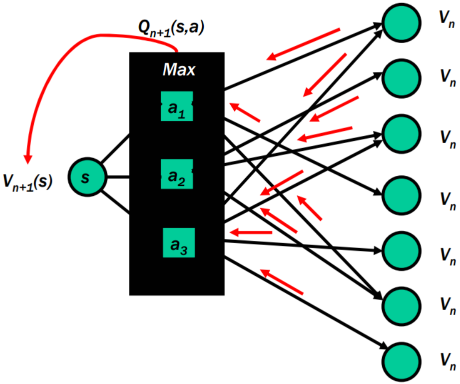
- $R(s)$ - immediate reward for state s
- $U(s')$ - utility of the next state s'
- assume that the agent chooses the optimal action

Value Iteration

- $\forall s, U(s) = 0$
- for each iteration i

$$\forall s, U_{i+1}(s) \leftarrow R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s' \mid s, a) U_i(s')$$

Value Iteration



Outline

Policy Iteration

- initial policy π_0
- for each iteration i
  - **policy evaluation**

$$\forall s \in S, U_{i+1} \leftarrow R(s) + \gamma \sum_{s'} P(s' \mid s, \pi_i(s)) U_i(s')$$

  - **policy improvement**

$$\forall s \in S, \pi_{i+1}(s) \leftarrow \arg \max_{a \in A(s)} \sum_{s'} P(s' \mid s, a) U_{i+1}(s')$$