Decision Theory: Value Iteration

CPSC 322 – Decision Theory 4

Textbook §9.5

Lecture Overview

1. Recap
2. Policies
3. Value Iteration

Value of Information and Control

Definition (Value of Information)

The value of information X for decision D is the utility of the the network with an arc from X to D minus the utility of the network without the arc.

Definition (Value of Control)

The value of control of a variable X is the value of the network when you make X a decision variable minus the value of the network when X is a random variable.

Markov Decision Processes

Definition (Markov Decision Process)

A Markov Decision Process (MDP) is a 5-tuple $\langle S, A, P, R, s_0 \rangle$, where each element is defined as follows:

- S: a set of states.
- A: a set of actions.
- $P(S_{t+1}|S_t, A_t)$: the dynamics.
- $R(S_t, A_t, S_{t+1})$: 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'.
- s_0: the initial state.

Rewards and Values

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:

$$V = \sum_{i=1}^{\infty} \gamma^{i-1} r_i$$

  - γ is the discount factor, $0 \leq \gamma \leq 1$

Lecture Overview

1. Recap
2. Policies
3. Value Iteration

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 π 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.
- Note: this means that although the environment is random, there's no benefit for the *agent* to randomize.

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

$$\begin{aligned} V^{\pi}(s) &= Q^{\pi}(s,\pi(s)) \\ Q^{\pi}(s,a) &= \sum_{s'} P(s'|a,s)\left(r(s,a,s') + \gamma V^{\pi}(s')\right) \end{aligned}$$

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:

$$\begin{aligned}
Q^*(s,a) &= \sum_{s'} P(s'|a,s)\left(r(s,a,s') + \gamma V^*(s')\right) \\
V^*(s) &= \max_a Q^*(s,a) \\
\pi^*(s) &= \arg\max_a Q^*(s,a)
\end{aligned}$$

Lecture Overview

1. Recap
2. 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:

$$
\begin{aligned}
Q_{i+1}(s,a) &= \sum_{s'} P(s'|a,s)\left(r(s,a,s') + \gamma V_i(s')\right) \\
V_{i+1}(s) &= \max_a Q_{i+1}(s,a)
\end{aligned}
$$

- 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, θ)
inputs:
  P is state transition function specifying $P(s'|a, s)$
  r is a reward function $R(s, a, s')$
  θ 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
      $V_k[s] = \max_a \sum_{s'} P(s'|a, s)(R(s, a, s') + \gamma V_{k-1}[s'])$
    if $\forall s \; |V_k(s) - V_{k-1}(s)| < \theta$
      for each state s
        $\pi(s) = \arg\max_a \sum_{s'} P(s'|a, s)(R(s, a, s') + \gamma V_{k-1}[s'])$
      return π, V_k
end

Value Iteration Example: Gridworld

See
`http://www.cs.ubc.ca/spider/poole/demos/mdp/vi.html`.