RL: Basics

Value Iteration

Marius Lindauer

Leibniz Universität Hannover

Automated Machine Learning Hannover

AutoML.org

MDP: Computing Optimal Policy and Optimal Value

- Policy iteration computes optimal value and policy
- Value iteration is another technique
  - Idea: Maintain optimal value of starting in a state s if have a finite number of steps k left in the episode
  - Iterate to consider longer and longer episode

Bellman Equation and Bellman Backup Operators

- Value function of a policy must satisfy the Bellman equation

$$V^{\pi}(s) = R^{\pi}(s) + \gamma \sum_{s' \in S} P^{\pi}(s' \mid s) V^{\pi}(s')$$

- Bellman backup operator
  - Applied to a value function
  - Returns a new value function
  - Improves the value if possible

$$BV(s) = \max_{a} [R(s,a) + \gamma \sum_{s' \in S} p(s' \mid s,a) V(s')]$$

  - BV yields a value function over all states s
  - Note: Read B as an operator applied to V

Automated Machine Learning Hannover

AutoML.org

Value Iteration (VI)

- Set $k = 1$
- Initialize $V_0(s) = 0$ for all states s
- Loop until convergence
  - For each state s

$$V_{k+1}(s) = \max_{a \in A} R(s, a) + \gamma \sum_{s' \in S} P(s' \mid s, a) V_k(s')$$

  - View as Bellmann backup on value function

$$V_{k+1} = BV_k$$

$$\pi_{k+1}(s) \in \arg\max_{a \in A} R(s, a) + \gamma \sum_{s' \in S} P(s' \mid s, a) V_k(s')$$

Policy Iteration as Bellman Operations

- Bellman backup operator B^π for a particular policy is defined as

$$B^\pi V(s) = R^\pi(s) + \gamma \sum_{s' \in S} P^\pi(s' \mid s) V(s')$$

- Policy evaluation amounts to computing the fixed point of B^π
- To do policy evaluation, repeatedly apply operator until V stops changing

$$V^\pi = B^\pi B^\pi B^\pi B^\pi \dots B^\pi V$$

Policy Iteration as Bellman Operations

- Bellman backup operator B^π for a particular policy is defined as

$$B^\pi V(s) = R^\pi(s) + \gamma \sum_{s' \in S} P^\pi(s' \mid s) V(s')$$

- To do policy improvement

$$\pi_{k+1}(s) \in \arg\max_{a \in A} R(s,a) + \gamma \sum_{s' \in S} P(s' \mid s, a) V^{\pi_k}(s')$$

Automated Machine Learning Hannover

Going back to Value Iteration (VI)

- Set $k = 1$
- Initialize $V_0(s) = 0$ for all states s
- Loop until convergence
  - For each state s

$$V_{k+1}(s) = \max_{a \in A} R(s,a) + \gamma \sum_{s' \in S} P(s' \mid s, a) V_k(s')$$

  - Equivalent in Bellman backup notation

$$V_{k+1} = BV_k$$

  - To extract optimal policy if can act for $k+1$ more steps

$$\pi_{k+1}(s) \in \arg\max_{a \in A} R(s,a) + \gamma \sum_{s' \in S} P(s' \mid s, a) V_{k+1}(s')$$

AutoML.org Automated Machine Learning Hannover

Going back to Value Iteration (VI)

- Yes, if discount factor $\gamma < 1$, or end up in a terminal state with probability 1
- Bellman backup is a contraction if discount factor, $\gamma < 1$
- If apply it to two different value functions, distance between value functions shrinks after applying Bellman equation to each
- (Skip proof)

What you should know

- Define MP, MRP, MDP, Bellman operator,Q-value, policy
- Be able to implement
  - Value Iteration
  - Policy Iteration
- Which policy evaluation methods require the Markov assumption?