Policy Iteration

MICHAEL WOLLOWSKI

Policies, policies, policies

After the next session, we will have seen three different ways to generated policies:

- Value Iteration: Recalculate utilities until no significant changes, then “read off” optimal policy.
- Policy Iteration: Start with a random policy and improve it until no changes.
- Q-Learning: Start with nothing and wing it.

Recap: Value Iteration

In Value Iteration, we determine the values of each state through an iterative process.

- We start with values zero for all states except for the absorbing states.
- We iterate until no significant changes occur
- We return the utility matrix, containing the values for each state.

Based on the utility matrix, we then determine an optimal policy by:

- looking at all possible actions and
- Using the stochastic transition function
- Selecting the action that leads to he highest utility value

Policy Evaluation and Improvement

Policy Evaluation: Given a policy *p*, calculate $U_i = U^{\pi i}$, the utility of each state if π_i where to be executed.

$$U_i(s) = R(s) + \gamma \sum_{s'} P(s' \mid s, \pi_i(s))\, U_i(s')$$

Policy Improvement: Calculate a new maximum expected policy π_{i+1}, using one-step look-ahead based on U_i

$$\textbf{if} \max_{a \in A(s)} \sum_{s'} P(s' \mid s, a)\, U[s'] > \sum_{s'} P(s' \mid s, \pi[s])\, U[s']$$

$$\textbf{then}\ \pi[s] := \operatorname{argmax}_{a \in A(s)} \sum_{s'} P(s' \mid s, a)\, U[s']$$

Policy Iteration

function POLICY-ITERATION(mdp) **returns** a policy
  inputs: mdp, an MDP with states S, actions $A(s)$, transition model $P(s' \mid s, a)$
  local variables: U, a vector of utilities for states in S, initially zero
    π, a policy vector indexed by state, initially random

  repeat
    $U \leftarrow$ POLICY-EVALUATION(π, U, mdp)
    $unchanged? \leftarrow$ true
    for each state s **in** S **do**
      if $\max_{a \in A(s)} \sum_{s'} P(s' \mid s, a)\ U[s'] > \sum_{s'} P(s' \mid s, \pi[s])\ U[s']$ **then do**
        $\pi[s] \leftarrow \operatorname{argmax}_{a \in A(s)} \sum_{s'} P(s' \mid s, a)\ U[s']$
        $unchanged? \leftarrow$ false
  until $unchanged?$
  return π

Algorithm source: Russell and Norvig: AIMA 2nd Ed.

Class Exercise

Consider the following partial policy:

3			↑	+1
2		W		-1
1	*Start*			
	1	2	3	4

Assume that the values of the states are either -0.04 or as indicated.

Focus on state s = <3, 3> and calculate utility as well as the new policy for one iteration.

Assume γ to be 0.9.