



# Dynamic Programming

Rupam Mahmood

February 5, 2020


# Worksheet question

A deterministic policy $\pi(s)$ outputs an action $a \in \mathcal{A} = \{a_1, a_2, \ldots, a_k\}$ directly. More generally, a policy $\pi(\cdot|s)$ outputs the probabilities for all actions: $\pi(\cdot|s) = [\pi(a_1|s), \pi(a_2|s), \ldots, \pi(a_k|s)$. How can you write a deterministic policy in this form? Let $\pi(s) = a_i$ and define $\pi(\cdot|s)$.

# Worksheet question

In iterative policy evaluation, we seek to find the value function for a policy $\pi$ by applying the Bellman equation many times to generate a sequence of value functions $v_k$ that will eventually converge to the true value function $v_\pi$. How can we modify the update below to generate a sequence of action value functions $q_k$?

$$v_{k+1}(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) \left[r + \gamma v_k(s')\right]$$

# Worksheet question

The policy iteration algorithm on page 80 has a subtle bug in that it may never terminate if the policy continually switches between two or more policies that are equally good. This is ok for pedagogy, but not for actual use. Modify the pseudocode so that convergence is guaranteed. Note that there is more than one approach to solve this problem.

**Policy Iteration (using iterative policy evaluation) for estimating $\pi \approx \pi_*$**

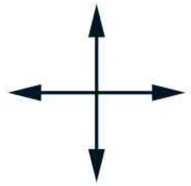
1. Initialization
   $V(s) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation
   Loop:
      $\Delta \leftarrow 0$
      Loop for each $s \in \mathcal{S}$:
         $v \leftarrow V(s)$
         $V(s) \leftarrow \sum_{s',r} p(s', r \mid s, \pi(s)) \big[r + \gamma V(s')\big]$
         $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
   until $\Delta < \theta$ (a small positive number determining the accuracy of estimation)

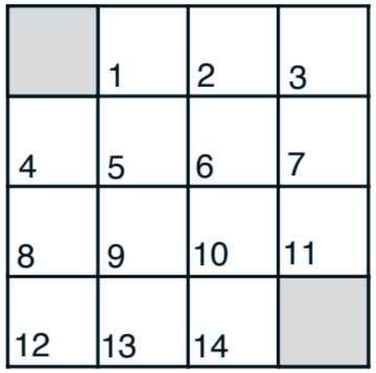
3. Policy Improvement
   *policy-stable* $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
      *old-action* $\leftarrow \pi(s)$
      $\pi(s) \leftarrow \arg\max_a \sum_{s',r} p(s', r \mid s, a) \big[r + \gamma V(s')\big]$
      If *old-action* $\neq \pi(s)$, then *policy-stable* $\leftarrow$ *false*
   If *policy-stable*, then stop and return $V \approx v_*$ and $\pi \approx \pi_*$; else go to 2

# Worksheet question

(*Exercise 4.1 S&B)* Consider the 4x4 gridworld below, where actions that would take the agent off the grid leave the state unchanged. The task is episodic with $\gamma = 1$ and the terminal states are the shaded blocks. Using the precomputed values for the equiprobable policy below, what is $q_\pi(11, \text{down})$? What is $q_\pi(7, \text{down})$?



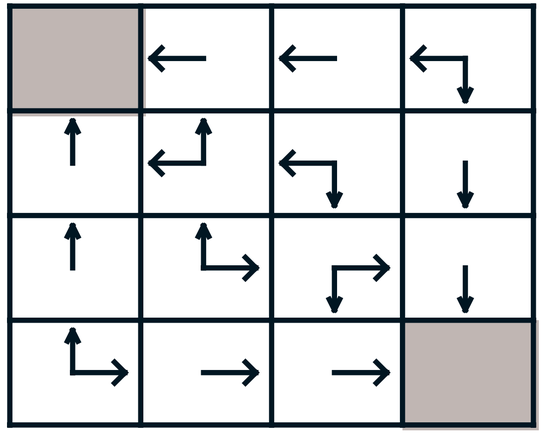
actions




$R_t = -1$
on all transitions

$k = \infty$

| | | | |
|---|---|---|---|
| 0.0 | -14. | -20. | -22. |
| -14. | -18. | -20. | -20. |
| -20. | -20. | -18. | -14. |
| -22. | -20. | -14. | 0.0 |



# Worksheet question

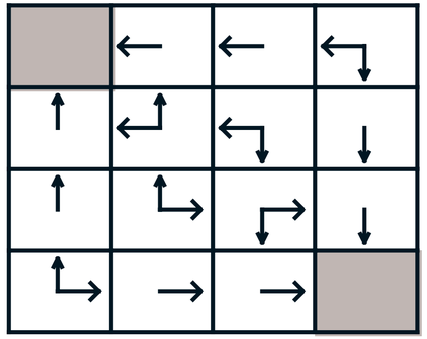
(*Exercise 4.1 from S&B*) Suppose in the above gridworld where a new state 15 is added to the gridworld just below state 13, and its actions, left, up, right, and dowm, take the agent to the states 12, 13, 14, and 15, respectively. Assume that the transitions from the original states are unchanged. What, then is, $v_\pi(15)$ for the equiprobable random policy? Now suppose the dynamics of state 13 are also changed, such that action down from state 13 takes the agent to the new state 15. What is $v_\pi(15)$ for the equiprobable random policy in this case?


| | | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 |
| 12 | 13 | 14 | |

$R_t = -1$
on all transitions

$k = \infty$

| | | | |
|---|---|---|---|
| 0.0 | -14. | -20. | -22. |
| -14. | -18. | -20. | -20. |
| -20. | -20. | -18. | -14. |
| -22. | -20. | -14. | 0.0 |



# Demo of an MDP in action