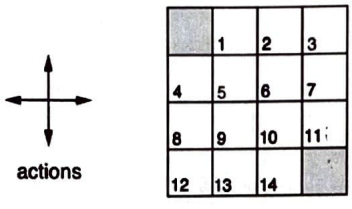
Name: Alfred Cuevas
Student No.: 2020-12788

# Exercise 2

## Data Science & Reinforcement Learning

Spring 2023

1. Consider the 4×4 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.



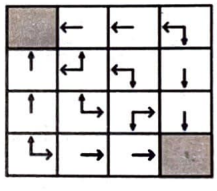

$R_t = -1$ on all transitions

0.25

Using the precomputed state values ($v_\pi$) for the equiprobable (uniform random) policy below, what is $q_\pi(11, down)$? What is $q_\pi(7, down)$?

$k = \infty$

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



$v_\pi(11, down) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)[r + \gamma v_k(s')]$

$= 0.25(-1 + (1)(0)) = -\frac{1}{4}$

- because $r = -1$

$q_\pi(11, down) = -1 + 0 = -1$

$v_\pi(7, down) = -14$

$= 0.25(-1 + (1)(-14)) = -\frac{15}{4}$

$q_\pi(7, down) = -1 - 14 = -15$

2. 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 $q_k$?

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

$q(s,a) = E[R_{t+1} + \gamma V(S_{t+1})]$

$q_{k+1}(s,a) = \sum_{s',r} P(s', r|s, a)[r + \gamma v_k(s')] = \sum_{s',r} P(s', r|s, a)[r + \gamma \sum_a \pi(a|s) q_k(s', a)]$

3. A gambler has the opportunity to make bets on the outcomes of a sequence of coin flips. If the coin comes up heads, she wins as many dollars as she has staked on that flip; if it is tails, she loses her stake. The game ends when the gambler wins by reaching her goal of \$100, or loses by running out of money. On each flip, the gambler must decide what portion of her capital to stake, in integer numbers of dollars. This problem can be formulated as an undiscounted, episodic, finite MDP. The state is the gambler's capital, $s \in \{1, 2, ..., 99\}$ and the actions are stakes, $a \in \{0, 1, ..., \min(s, 100-s)\}$. The reward is +1 when reaching the goal of \$100 and zero on all other transitions. The probability of seeing heads is $p_h = 0.4$.

   (a) What does the value of a state mean in this problem? For example, in a gridworld where the value of 1 per step, the value represents the expected number of steps to goal. What does the value of state mean in the gambler's problem? Think about the minimum and maximum possible values, and think about the values of state 50 (which is 0.4) and state 99 (which is near 0.95).

It would be the probability of winning at each state from a given initial state using greedy policy, because till s(50) we match all heads.

state 50: $0,4 \times 1 + 0,6 \times 0 = 0,4$

$0,4 \times + 0,4( \quad ) \approx 0,95$

   (b) Modify the pseudocode for value iteration to more efficiently solve this specific problem, by exploiting your knowledge of the dynamics. *Hint: Not all states transition to every other state. For example, can you transition from state 1 to state 99?*

**Value Iteration, for estimating $\pi \approx \pi_*$**

**Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation**
**Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$**

**Loop:**
  $\Delta \leftarrow 0$
  **Loop for each $s \in \mathcal{S}$:**
    $v \leftarrow V(s)$
    $V(s) \leftarrow \max_a \sum_{s',r} p(s', r|s, a)[r + \gamma V(s')]$
    $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
**until $\Delta < \theta$**

**Output a deterministic policy, $\pi \approx \pi_*$, such that**
$\pi(s) = \text{argmax}_a \sum_{s',r} p(s', r|s, a)[r + \gamma V(s')]$

if tails $\rightarrow V = 0$

$$V(s) = \max_a \sum_{s',r} p(s',r|s,a)[r + \gamma V(s')]$$

$$V(s) = \max_a \left( 0,4\,[r(s,a) + \gamma V(s+a)] + 0,6\,[r(s-a) + \gamma V(s-a)] \right)$$

$$\pi(s) = \arg\max_a \left( 0,4\,[r(s,a) + \gamma V(s+a)] + 0,6\,[r_{(s-a)} + \gamma V(s-a)] \right).$$

4. The policy iteration algorithm (on page 80 of the textbook) 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 teaching purposes, 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 $s \in \mathcal{S}$:
   $v \leftarrow V(s)$
   $V(s) \leftarrow \sum_{s',r} p(s',r|s,\pi(s))[r + \gamma V(s')]$
   $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
   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* $\leftarrow \pi(s)$
   $\pi(s) \leftarrow \arg\max_a \sum_{s',r} p(s',r|s,a)[r + \gamma V(s')]$
   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

whole block → because if old-action $\neq \pi(s)$ only considers Policy
we should evaluate the action value to see if it's optimal action-value won't change
→ calculate $V_\pi(s)$ and $V_{\pi(old)}$
old_action $\leftarrow \pi(s)$ and old-value $\leftarrow V(s)$

Same level

5. How would policy iteration be defined for action values? Give a complete algorithm for computing $q^*$, analogous to the policy iteration for computing $v^*$ (shown above).

if: if $V_\pi(s) > V_{\pi(old)}(s)$:
$\pi(s) \leftarrow$ new-action
policy-stable $\leftarrow$ false

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

2. Policy evaluation
Loop:
$\Delta \leftarrow 0$
Loop for each $s \in S$:
$q \leftarrow Q_\pi(s,a)$
$Q_\pi(s,a) \leftarrow \sum_{s',r} P(s',r|s,a)[r + \gamma q(s', \pi(s'))]$
$\Delta \leftarrow \max(\Delta, |q - Q(s,a)|)$
until $\Delta < \theta$

3. Policy improvement
policy-stable $\leftarrow$ true
for each $s \in S$:
old-action $\leftarrow \pi(s)$
$\pi(s) \leftarrow \arg\max_a q(s,a)$
if old-action $\neq \pi(s)$:
policy-stable $\leftarrow$ false
if policy-stable then
stop and return $q = q_*$
and $\pi \approx \pi_*$; else go to 2