Reinforcement Learning: Tutorial

Clemens Heitzinger

March 20, 2025

Contents

1 Tutorial on 27 March 2025 (18 exercises)

1. **ENV/EX** – *Think of application.*
   Think of a (preferably creative) application of reinforcement learning. Specify the states, actions, and rewards as well as what is needed to satisfy the Markov property.

2. **ENV/COUNTEREX** – *Goal-directed learning task that is not an MDP.*
   Try to find a goal-directed learning task that cannot be represented by a Markov decision process.

3. **AS** – *ϵ-greedy action selection.*
   Assume that ϵ-greedy action selection is used.

   (a) Suppose $|\mathcal{A}| = 4$ and $\epsilon = 0.2$. When using ϵ-greedy action selection, what is the probability that the greedy action is selected? 0.8 + 0.2*1/4 = 0.85

   (b) Which value of ϵ would achieve a probability of 70% of selecting the greedy action? 0.4 1-eps + eps*1/4 = 0.75

   (c) Generalize the formula for calculating the probability of selecting the greedy action in ϵ-greedy action selection for any $|\mathcal{A}|$ and any ϵ.
   1-epsilon + epsilon/IAI

4. **STS/H** – *Harmonic step sizes.*
   Show that the step sizes

$$\alpha_n := \frac{1}{an+b}, \qquad a, b \in \mathbb{R},$$

   (where $a \in \mathbb{R}^+$ and $b \in \mathbb{R}$ are chosen such that $an + b \neq 0$) satisfy the convergence conditions

$$\sum_{n=1}^{\infty} \alpha_n = \infty, \qquad \sum_{n=1}^{\infty} \alpha_n^2 < \infty.$$

5. **STS/U** – *Unbiased step sizes.*
   We use the iteration

$$Q_1 \in \mathbb{R},$$
$$Q_{n+1} := Q_n + \alpha_n(R_n - Q_n), \qquad n \geq 1,$$

   to estimate Q_n using R_n, where

$$\alpha_n := \frac{\alpha}{\beta_n}, \qquad \alpha \in (0,1), \quad n \geq 1,$$

   and

$$\beta_0 := 0,$$
$$\beta_n := \beta_{n-1} + \alpha(1 - \beta_{n-1}), \qquad n \geq 1.$$

Show that the iteration for Q_n above yields an exponential recency-weighted average *without initial bias* (i.e., the Q_n do not depend on the initial value Q_1).

6. **MAB/EPS** – *Multi-armed bandits with ϵ-greedy action selection (programming).*
   You play against a 10-armed bandit, where at the beginning of each episode the true value $q_*(a)$, $a \in \{1, \dots, 10\}$, of each of the 10 actions is chosen to be normally distributed with mean zero and unit variance. The rewards after choosing action/bandit a are normally distributed with mean $q_*(a)$ and unit variance. Using the simple bandit algorithm and ϵ-greedy action selection, you have 1000 time steps or tries in each episode to maximize the average reward starting from zero knowledge about the bandits.

   Which value of ϵ maximizes the average reward? Which value of ϵ maximizes the percentage of optimal actions taken?

7. **MAB/UCB** – *Multi-armed bandits with upper-confidence-bound action selection (programming).*
   This exercise is the same as in Exercise MAB/EPS, but now the actions

   $$A_t := \arg\max_a \left(Q_t(a) + c\sqrt{\frac{\ln t}{N_t(a)}} \right)$$

   are selected according to the upper-confidence bound.

   Which value of c yields the largest average reward?

8. **MAB/SOFTMAX** – *Multi-armed bandits with soft-max action selection (programming).*
   This exercise is the same as Exercise MAB/EPS, but now the actions $A_t \in \mathcal{A} = \{1, \dots, |\mathcal{A}|\}$ are selected with probability

   $$\mathbb{P}[a] = \frac{\exp(Q_t(a)/\tau)}{\sum_{i=1}^{|\mathcal{A}|} \exp(Q_t(i)/\tau)},$$

   where the parameter τ is called the temperature. This probability distribution is called the soft-max or Boltzmann distribution.

   What are the effects of low and high temperatures, i.e., how does the temperature influence the probability distribution all else being equal? Which value of τ yields the largest average reward?

9. **MDP/G1** – *Returns and episodes.*
   Suppose $\gamma := 1/2$ and the rewards $R_1 := 1$, $R_2 := -1$, $R_3 := 2$, $R_4 := -1$, and $R_5 := 2$ are received in an episode with length $T := 5$. What are $G_0, \dots, G_5$?

10. MDP/G2 – *Returns and episodes.*
Suppose $\gamma := 0.9$ and the reward sequence starts with $R_1 := -1$ and $R_2 := 2$ and is followed by an infinite sequence of 1s. What are G_0, G_1, and G_2?

11. MDP/V – *Equation for v_π.*
Give an equation for v_π in terms of q_π and π.

12. MDP/Q – *Equation for q_π.*
Give an equation for q_π in terms of v_π and the four-argument p.

13. MDP/RET – *Change of return.*
In episodic tasks and in continuing tasks, how does the return G_t change if a constant c is added to all rewards R_t?

14. MDP/BELLMAN/QPI – *Bellman equation for q_π.*
Analogous to the derivation of the Bellman equation for v_π, derive the Bellman equation for q_π.

15. MDP/VSTAR – *Equation for v_*.*
Give an equation for v_* in terms of q_*.

16. MDP/QSTAR – *Equation for q_*.*
Give an equation for q_* in terms of v_* and the four-argument p.

17. MDP/PISTAR/VSTAR – *Equation for π_*.*
Give an equation for π_* in terms of q_*.

18. MDP/PISTAR/QSTAR – *Equation for π_*.*
Give an equation for π_* in terms of v_* and the four-argument p.

2 Tutorial on 3 April 2025 (7 exercises)

1. **DP/BANACH** – *Formulate Banach fixed-point theorem.*
   Formulate the Banach fixed-point theorem after defining all relevant terms.

2. **DP/BANACH/PROOF** – *Prove Banach fixed-point theorem.*
   Prove the Banach fixed-point theorem.

3. **DP/UPDATE/Q** – *Update rule for q_π.*
   Using the Bellman equation for q_π (see Exercise MDP/BELLMAN/QPI), find an update rule for the approximation q_{k+1} of q_π (in terms of q_k, π, and p) analogous to the update rule for v_{k+1}.

4. **GW/SIMPLE** – *Simple 4×4 grid world (programming).*
   Implement a 4×4 grid world with two terminal states in the upper left corner and lower right corners (resulting in 14 non-terminal states). The four actions $\mathcal{A} = \{\text{up}, \text{down}, \text{left}, \text{right}\}$ act deterministically, the discount factor is $\gamma = 1$, and the reward is always equal to -1. Ensure that a maximum number of time steps can be specified.

5. **DP/POLICY/EVAL** – *Iterative policy evaluation (programming).*
   Implement iterative policy evaluation and use it to estimate v_π for the grid world in Exercise GW/SIMPLE, where π is the equiprobable random policy.

6. **DP/POLICY/ITER** – *Policy iteration (programming).*
   Implement policy iteration and use it to estimate π_* for the grid world in Exercise GW/SIMPLE.

7. **DP/VALUE/ITER** – *Value iteration (programming).*
   Implement value iteration and use it to estimate π_* for the grid world in Exercise GW/SIMPLE.

References

[1] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: an Introduction.* The MIT Press, 2nd edition edition, 2018.

[2] Mohammad Ghavamzadeh, Hilbert J. Kappen, Mohammad G. Azar, and Rémi Munos. Speedy Q-learning. In J. Shawe-Taylor, R.S. Zemel, P.L. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24 (NIPS 2011)*, pages 2411–2419. Curran Associates, Inc., 2011.