The Johns Hopkins University
Department of Electrical and Computer Engineering

Foundations of Reinforcement Learning

Problem Set #3

Homework Rules:

- Whenever a proof is required, please provide a detailed justification of every step.
- Whenever Python/Matlab simulations are required, please attach a copy of the code. The code should be legible by an experienced reader.
- **Reference book**: Reinforcement Learning: An Introduction, 2nd edition, by Richard S. Sutton and Andrew G. Barto.
- Unless explicitly stated otherwise, exercises are expected to be solved by yourself using only Blackboard resources and your own lecture notes. Verbal and piazza discussions are acceptable, provided that do not involve any explicit writing of the solution.

Problems.

1. For the multi-arms bandit problem we discussed in the class. Suppose that we get return G_n at n-th time we do action a, and $\mathbb{E}G_n = r,\ n = 1, 2, \cdots$.

   Let Q_{n+1} be our estimates of r after we do action a the n-th time, and we have the following update rule

   $$Q_{n+1} = Q_n + \alpha_n(G_n - Q_n)\,,\ Q_1 = 0\,.$$

   We define $V_n = \mathbb{E}\left[(Q_n - r)^2\right]$.

   (a) (Decreasing step size) Let $\alpha_n = \frac{1}{n}$, show that
   - i. (5 points) $Q_{n+1} = \frac{1}{n}\sum_{i=1}^{n} G_i,\ n = 1, 2, \cdots$,
   - ii. (5 points) $\lim_{n\to\infty} V_n = 0\,.$

   (b) (Constant step size) Let $\alpha_n = \alpha, 0 < \alpha < 2$, show that
   - i. (10 points) $V_{n+1} = (1-\alpha)^2 V_n + \alpha^2 \mathrm{Var}[G_n]$, where $\mathrm{Var}[G_n] = \mathbb{E}\left[(G_n - r)^2\right]$
   - ii. (15 points) $\lim_{n\to\infty}\left|V_{n+1} - \frac{\alpha}{2-\alpha}\mathrm{Var}[G_n]\right| = 0.$

2. (25 points) Answer Sutton&Barto Exercise 2.7

Exercise 2.7: Unbiased Constant-Step-Size Trick In most of this chapter we have used sample averages to estimate action values because sample averages do not produce the initial bias that constant step sizes do (see the analysis leading to (2.6)). However, sample averages are not a completely satisfactory solution because they may perform poorly on nonstationary problems. Is it possible to avoid the bias of constant step sizes while retaining their advantages on nonstationary problems? One way is to use a step size of

$$\beta_n \doteq \alpha/\bar{o}_n, \tag{2.8}$$

to process the nth reward for a particular action, where $\alpha > 0$ is a conventional constant step size, and $\bar{o}_n$ is a trace of one that starts at 0:

$$\bar{o}_n \doteq \bar{o}_{n-1} + \alpha(1 - \bar{o}_{n-1}), \quad \text{for } n \geq 0, \quad \text{with } \bar{o}_0 \doteq 0. \tag{2.9}$$

Carry out an analysis like that in (2.6) to show that Q_n is an exponential recency-weighted average *without initial bias*. □

Supplement contents in the textbook:

2.5 Tracking a Nonstationary Problem

The averaging methods discussed so far are appropriate for stationary bandit problems, that is, for bandit problems in which the reward probabilities do not change over time. As noted earlier, we often encounter reinforcement learning problems that are effectively nonstationary. In such cases it makes sense to give more weight to recent rewards than to long-past rewards. One of the most popular ways of doing this is to use a constant step-size parameter. For example, the incremental update rule (2.3) for updating an average Q_n of the $n-1$ past rewards is modified to be

$$Q_{n+1} \doteq Q_n + \alpha\Big[R_n - Q_n\Big], \tag{2.5}$$

where the step-size parameter $\alpha \in (0, 1]$ is constant. This results in Q_{n+1} being a weighted average of past rewards and the initial estimate Q_1:

$$\begin{aligned}
Q_{n+1} &= Q_n + \alpha\Big[R_n - Q_n\Big] \\
&= \alpha R_n + (1-\alpha)Q_n \\
&= \alpha R_n + (1-\alpha)\left[\alpha R_{n-1} + (1-\alpha)Q_{n-1}\right] \\
&= \alpha R_n + (1-\alpha)\alpha R_{n-1} + (1-\alpha)^2 Q_{n-1} \\
&= \alpha R_n + (1-\alpha)\alpha R_{n-1} + (1-\alpha)^2\alpha R_{n-2} + \\
&\qquad\qquad \cdots + (1-\alpha)^{n-1}\alpha R_1 + (1-\alpha)^n Q_1 \\
&= (1-\alpha)^n Q_1 + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i.
\end{aligned} \tag{2.6}$$