Notes on *Reinforcement Learning: An Introduction* by Sutton & Barto

Luke Renchik

July 1, 2024

Introduction

- **Authors:** Richard S. Sutton and Andrew G. Barto
- **Book Title:** Reinforcement Learning: An Introduction
- **Edition:** Second Edition
- **Publication Year:** 2018
- **Purpose:** Provide a comprehensive introduction to the field of reinforcement learning.

Chapter 2.6-2.9: Multi-Armed Bandits

Important Notes

- Chapter 2.6-2.9

  2.6 - Optimistic Initial Values

  Method where initial values are instantiated at some value greater than actual rewards so algorithm explores all signals early.

  2.7 - Upper-Confidence-Bound Action Selection

  Utilizes a method that weights action selection based on how many times the action has been selected previous compared to the number of time-steps on a logarithmic scale. Not practical for the full reinforcement learning problem.

  2.8 - Gradient Bandit Algorithms

  Selects signals based on their performance relative to expected values, i.e. those who are trending positively become favored much more heavily.

  2.9 - Associative Search (Contextual Bandits)

  Investigating situations where there are many possible states and needs for the agent.

Definitions

- **Associative Search:** Decide best action to take based on state of machine. This is an intermediary between the k-armed bandit problem and the full reinforcement learning problem. The reinforcement learning problem has the ability for the actions to influence the next situation in addition to the current one.

Exercises

- 2.10 - Suppose you face a 2-armed bandit task whose true action values change randomly from time step to time step. Specifically, suppose that, for any time step, the true values of actions 1 and 2 are respectively 10 and 20 with probability of 0.5 (case A), and 90 and 80 with probability 0.5 (case B). If you are not able to tell which case you face at any step, what is the best expected reward you can achieve and how should you behave

to achieve it?

The best reward can be found by:
$\sum_0^i$**(probability**$_i$ *** value**$_i$**)**
for the different cases then comparing the total, in this case both choices are equivalent.

Now suppose that on each step you are told whether you are facing case A or case B. This is an associative search task. What is the best expected reward you can achieve in this task, and how should you behave to achieve it?

The best reward can be calculated by comparing the two values associated with the actions and picking the greater.

Personal Insights

- The four main algorithms used in reinforcement learning problems are the Sample Average, Optimistic Initial Value, Gradient Ascent, and Associative Search. It will be imperative to understand these, so I will implement on the k-armed testbed built previously. Implementations of the first three methods are available in the associated GitHub for these notes.

General Notation

Random Variables - Upper Case
Instantiated Variables - Lower Case
At time step t:
State = S_t
Action = A_t
Reward = R_t
The specific values would be s, a, r.
Value functions - lower case (e.g., v_π)
Tabular Estimates - upper case (e.g., Q_t(s,a))
Approximate value functions are deterministic functions of the random parameters and written in lowercase ($\hat{v}(s, \mathbf{w_t}) \approx v_\pi(s)$)
Bold Letters represent vectors
Uppercase Bold is used for matrices
Transition for State Function - $p(s', r|s, a)$
$\doteq$ - Equality relationship that is true by definition
$\propto$ - Proportional To
$\Pr\{X = x\}$ - Probability that a random variable X takes on the value x
$X \sim p$ - random variable X selected from distribution $p(x) \doteq \Pr\{X = x\}$
$\mathbb{E}[\mathbb{X}]$ - Expectation of a random variable X, i.e., $\mathbb{E}[\mathbb{X}] \doteq \sum_x p(x)x$)
$\ln x$ - Natural Log of X
e^x, exp(x) - The base of the natural log, e carried to the power x; $e^{\ln x}$ = x

$\mathbb{R}$ - Set of real numbers
$f : X \to y$ - Function f from elements of set X to elements of set y
$\leftarrow$ - Assignment
ϵ - Probability of takin a random action in an ϵ greedy policy.
α, β - step-size parameters
γ - Discount Rate Parameter λ - decay-rate parameter for eligibility traces
$\mathbb{I}_{predicate}$ = indicator function ($\mathbb{I}_{predicate} \doteq 1$ if the *predicate* is true, else 0)