Introduction to Reinforcement Learning

Part 9: Bandit Problems
N-armed Bandits

Sutton & Barto, Chapter 2

- The simplest problems with evaluative feedback (rewards)
- Evaluating actions vs. instructing by giving correct actions
- Supervised learning is instructive; optimization is evaluative

- Associative vs. Nonassociative:
  - Associative: inputs mapped to actions; learn the best action for each input
  - Nonassociative: “learn” (find) one best action

- n-armed bandit (at least how we treat it) is:
  - Nonassociative
  - Evaluative feedback
The n-Armed Bandit Problem

- Choose repeatedly from one of n actions; each choice is called a play
- After each play $A_t$, you get a reward $R_t$, where

$$E\{R_t\} = q^*(A_t)$$

These are unknown action values
Distribution of $R_t$ depends only on $A_t$

- Objective is to maximize the reward in the long term, e.g., over 1000 plays

To solve the n-armed bandit problem, you must explore a variety of actions and exploit the best of them
The Exploration/Exploitation Dilemma

- Suppose you form estimates
  \[ Q_t(a) \approx q_*(a) \]
  Action-value estimates

- The greedy action at \( t \) is \( A_t^* \)
  \[ A_t^* = \arg\max_a Q_t(a) \]

  \[ A_t = A_t^* \Rightarrow \text{exploitation} \]
  \[ A_t \neq A_t^* \Rightarrow \text{exploration} \]

- You can’t exploit all the time; you can’t explore all the time
- You can never stop exploring; but maybe you should always reduce exploring. Maybe not.
Action-Value Methods

Methods that adapt action-value estimates and nothing else, e.g.: suppose by the t-th play, action had been chosen $K_a$ times, producing rewards $R_1, R_2, \ldots, R_{K_a}$, then

$$Q_t(a) = \frac{R_1 + R_2 + \cdots + R_{K_a}}{K_a} \quad \text{“sample average”}$$

$$\lim_{K_a \to \infty} Q_t(a) = q^*(a)$$
\( \varepsilon \)-Greedy Action Selection

- Greedy action selection:
  \[
  A_t = A_t^* = \arg \max_a Q_t(a)
  \]

- \( \varepsilon \)-Greedy:
  \[
  A_t = \begin{cases} 
  A_t^* & \text{with probability } 1 - \varepsilon \\
  \text{a random action} & \text{with probability } \varepsilon
  \end{cases}
  \]

... the simplest way to balance exploration and exploitation
10-Armed Testbed

- $n = 10$ possible actions
- Each $q^*(a)$ is chosen randomly from a normal distribution: $N(0,1)$
- Each $R_t$ is also normal: $N(q^*(A_t),1)$
- 1000 plays
- Repeat the whole thing 2000 times and average the results
\( \varepsilon \)-Greedy Methods on the 10-Armed Testbed
Upper Confidence Bound (UCB) action selection

- A better way of reducing exploration over time
  - Compute bounds on the likely range of $q_*(a), \forall a$
  - Select the action with the largest upper bound
    - Adapt to both more plays to action $a$
    - and more plays on the other actions

$$A_t = \arg \max_a \left[ Q_t(a) + c \sqrt{\frac{\log t}{K_t(a)}} \right]$$
Incremental Implementation

Recall the sample average estimation method:

The average of the first $k$ rewards is (dropping the dependence on $a$):

$$Q_{k+1} = \frac{R_1 + R_2 + \cdots + R_k}{k}$$

Can we do this incrementally (without storing all the rewards)?

Could keep a running sum and count (and divide), or, equivalently:

$$Q_{k+1} = Q_k + \frac{1}{k}[R_k - Q_k]$$

This is a common form for update rules:

$$\text{NewEstimate} = \text{OldEstimate} + \text{StepSize}[\text{Target} - \text{OldEstimate}]$$
Choosing \( Q_k \) to be a sample average is appropriate in a stationary problem, i.e., when none of the \( q_*(a) \) change over time,

But not in a nonstationary problem.

Better in the nonstationary case is:

\[
Q_{k+1} = Q_k + \alpha [R_k - Q_k]
\]

for constant \( \alpha \), \( 0 < \alpha \leq 1 \)

\[
= (1 - \alpha)^k Q_0 + \sum_{i=1}^{k} \alpha (1 - \alpha)^{k-i} R_i
\]

exponential, recency-weighted average
Optimistic Initial Values

- All methods so far depend on $Q_0(a)$, i.e., they are biased.
- Suppose we initialize the action values optimistically, e.g., on the 10-armed testbed, use $Q_0(a) = 5$ for all $a$ and $\alpha = 0.1$. 

![Graph showing performance comparison between realistic, $\varepsilon$-greedy and optimistic, greedy policies.](image)
Conclusions

- These are all very simple methods
  - but they are complicated enough—we will build on them
  - we should understand them completely
- Ideas for improvements:
  - estimating uncertainties . . . interval estimation
  - UCB (Upper Confidence Bound)
  - approximating Bayes optimal solutions
  - Gittens indices
- The full RL problem offers some ideas for solution . . .