Date created: 2025-07-19-Sat

2.7 Upper Confidence Bound Action Selection

ϵ-greedy action selection encourages exploration by forcing non-greedy actions to by tried, but with no preference between them. Intuitively, we want to select non-greedy actions that are either:

- nearly greedy (i.e. really good but not best),
- very uncertain (i.e. has not been selected enough times).

In other words, it would be better to select among the non-greedy actions according to their potential for actually being optimal.

Upper-Confidence-Bound action selection

One effective way to take into account how close non-greedy estimates are to optimal and their uncertainties is the following scheme:

$$A_t := \underset{a}{\operatorname{argmax}} \left[Q_t(a) + c\sqrt{\frac{\ln t}{N_t(a)}} \right],$$

where

- $\ln t$ is the natural logarithm of t,
- $N_t(a)$ is the number of times action a has been selected prior to time t,
- $c > 0$ is a parameter that controls the degree of exploration.

Explanation

The square root term is a measure of the variance in the estimate of a's value. Thus, the quantity being maxed over could be thought of as an upper bound on the possible true value of a, with c determining the confidence level.

Each time a is selected, $N_t(a)$ increments $\implies$ variance term decreases, and each time a is not selected, t increments (but $N_t(a)$ does not) $\implies$ variance term increases. Additionally, the use of $\ln t$ means that the variance increases get smaller over time, but are unbounded.

The bottom line: all acions will eventually be selected, but actions with lower value estimates, or that have already been selected frequently, will be selected with decreasing frequency over time.

Example

As shown, UCB performs better than ϵ-greedy methods on 10-armed testbed. But UCB is more difficult than ϵ-greedy to extend beyond the bandit problem (e.g. dealing with nonstationary problems). Another difficulty of UCB is in dealing with large state spaces (high k), particularly when using function approximations (as well be introduced in Part II of the book). These limitations make UCB not always practical.

Exercises

2.8: UCB Spikes

In Figure 2.4 the UCB algorithm shows a distinct spike in performance on the 11th step. Why is this? Note that for your answer to be fully satisfactory it must explain both why the reward increases on the 11th step and why it decreases on the subsequent steps. Hint: if $c = 1$, then the spike is less prominent.

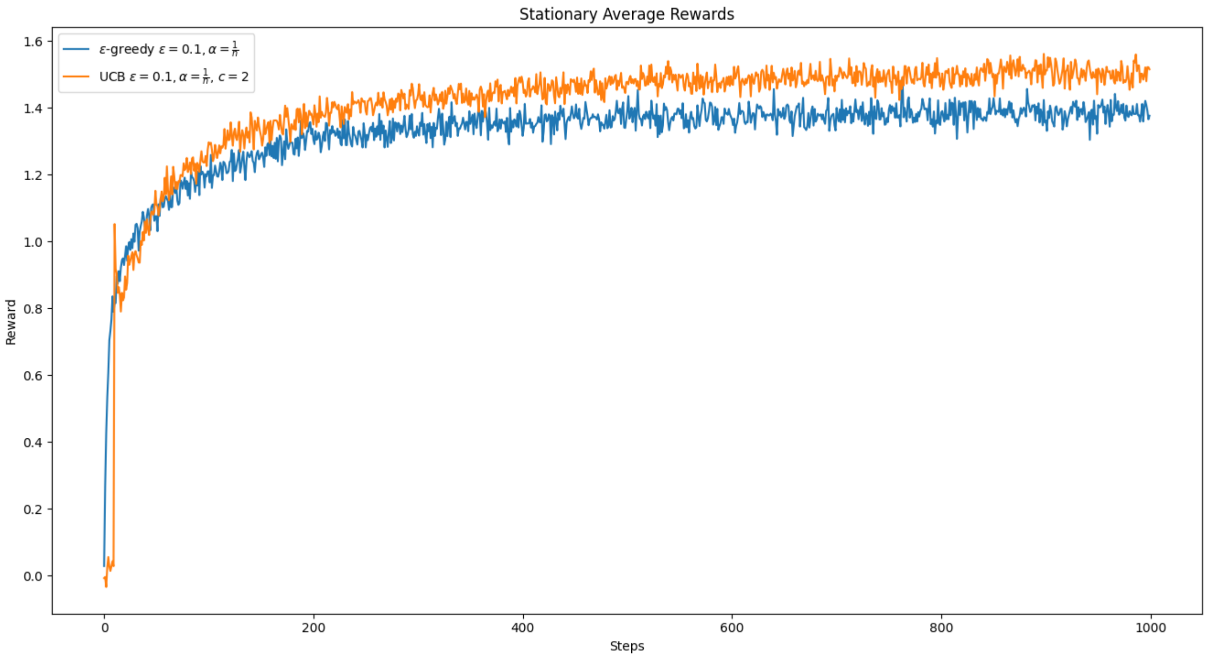

Figure 1: UCB vs. ϵ-greedy on 10-armed testbed

Solution Since UCB selects untried actions first (due to infinite uncertainty), each of the k actions is played exactly once in the first k steps. On step $k+1$, all actions have been tried once, so their uncertainty bonuses become equal:

$$c\sqrt{\frac{\ln(k+1)}{1}} = c\sqrt{\ln(k+1)}$$

As a result, the selection is purely based on the initial estimates $Q_k(a)$, which are noisy but unbiased. Over many runs, the best action (in expectation) is more likely to be selected at this step, creating a spike in average reward.

However, from step $k+2$ onward, actions start accumulating different counts again. Their uncertainty bonuses diverge, and UCB resumes exploring suboptimal actions due to its optimism. This causes the average reward to temporarily decrease after the spike.

The spike is especially noticeable with higher values of c, which amplify the early exploration bonus. When $c = 1$, the bonus is smaller, so the spike is less prominent.