Prof. Joschka Bödecker, Gabriel Kalweit

Reinforcement Learning
Sample Solution 8

1 UCB1

(a) Solve this task using pen and paper. Imagine a multi-armed bandit setting with three arms. Each arm has a bias $-+\frac{1}{2}$, $+\frac{1}{4}$ and $+\frac{1}{8}$, respectively. The Q-values for each arm are initialized by $1-b$, where b is the corresponding bias. The return for a pull is then $\text{clip}(b+u, 0, 1)$, where u is uniformly sampled noise from $[0, 1]$. Provide the first 10 iterations of the UCB1-algorithm based on the sampled noise array $[0.4, 0.7, 0.2, 0.3, 0.8, 0.5, 0.6, 0.1, 0.0, 0.9]$ for the returns. Estimate the Q-values by the mean.

Solution We have to calculate the UCB-values for each action and take the argmax of the sum in every step.

Iteration 0:

Q-values:
$Q_0(a_0) = 0.5$
$U_0(a_0) = \sqrt{\frac{2\log 1}{1}} = 0.0$
$\text{UCB}_0(a_0) = Q_0(a_0) + U_0(a_0) = 0.5$

$Q_0(a_1) = 0.75$
$U_0(a_1) = \sqrt{\frac{2\log 1}{1}} = 0.0$
$\text{UCB}_0(a_1) = Q_0(a_1) + U_0(a_1) = 0.75$

$Q_0(a_2) = 0.875$
$U_0(a_2) = \sqrt{\frac{2\log 1}{1}} = 0.0$
$\text{UCB}_0(a_2) = Q_0(a_2) + U_0(a_2) = 0.875$

Pulling the argmax: 2

Bias of bandit 2: 0.125
Random noise: 0.4
Return: 0.525

$Q_0(a_2) = 0.875 + \frac{1}{1}(0.525 - 0.875) = 0.525$

Iteration 1:

Q-values:
$Q_1(a_0) = 0.5$
$U_1(a_0) = \sqrt{\frac{2\log 2}{1}} = 1.177$
$\text{UCB}_1(a_0) = Q_1(a_0) + U_1(a_0) = 1.677$

$Q_1(a_1) = 0.75$
$U_1(a_1) = \sqrt{\frac{2\log 2}{1}} = 1.177$
$\text{UCB}_1(a_1) = Q_1(a_1) + U_1(a_1) = 1.927$

$Q_1(a_2) = 0.525$
$U_1(a_2) = \sqrt{\frac{2\log 2}{2}} = 0.833$
$\text{UCB}_1(a_2) = Q_1(a_2) + U_1(a_2) = 1.358$

Pulling the argmax: 1

Bias of bandit 1: 0.25
Random noise: 0.7
Return: 0.95

$Q_1(a_1) = 0.75 + \frac{1}{1}(0.95 - 0.75) = 0.95$

Iteration 2:

Q-values:
$Q_2(a_0) = 0.5$
$U_2(a_0) = \sqrt{\frac{2\log 3}{1}} = 1.482$
$\text{UCB}_2(a_0) = Q_2(a_0) + U_2(a_0) = 1.982$

$Q_2(a_1) = 0.95$
$U_2(a_1) = \sqrt{\frac{2\log 3}{2}} = 1.048$
$\text{UCB}_2(a_1) = Q_2(a_1) + U_2(a_1) = 1.998$

$Q_2(a_2) = 0.525$
$U_2(a_2) = \sqrt{\frac{2\log 3}{2}} = 1.048$
$\text{UCB}_2(a_2) = Q_2(a_2) + U_2(a_2) = 1.573$

Pulling the argmax: 1

Bias of bandit 1: 0.25

Random noise: 0.2
Return: 0.45

$Q_2(a_1) = 0.95 + \frac{1}{2}(0.45 - 0.95) = 0.7$

Iteration 3:

Q-values:
$Q_3(a_0) = 0.5$
$U_3(a_0) = \sqrt{\frac{2\log 4}{1}} = 1.665$
$\mathrm{UCB}_3(a_0) = Q_3(a_0) + U_3(a_0) = 2.165$

$Q_3(a_1) = 0.7$
$U_3(a_1) = \sqrt{\frac{2\log 4}{3}} = 0.961$
$\mathrm{UCB}_3(a_1) = Q_3(a_1) + U_3(a_1) = 1.661$

$Q_3(a_2) = 0.525$
$U_3(a_2) = \sqrt{\frac{2\log 4}{2}} = 1.177$
$\mathrm{UCB}_3(a_2) = Q_3(a_2) + U_3(a_2) = 1.702$

Pulling the argmax: 0

Bias of bandit 0: 0.5
Random noise: 0.3
Return: 0.8

$Q_3(a_0) = 0.5 + \frac{1}{1}(0.8 - 0.5) = 0.8$

Iteration 4:

Q-values:
$Q_4(a_0) = 0.8$
$U_4(a_0) = \sqrt{\frac{2\log 5}{2}} = 1.269$
$\mathrm{UCB}_4(a_0) = Q_4(a_0) + U_4(a_0) = 2.069$

$Q_4(a_1) = 0.7$
$U_4(a_1) = \sqrt{\frac{2\log 5}{3}} = 1.036$
$\mathrm{UCB}_4(a_1) = Q_4(a_1) + U_4(a_1) = 1.736$

$Q_4(a_2) = 0.525$
$U_4(a_2) = \sqrt{\frac{2\log 5}{2}} = 1.269$

$\text{UCB}_4(a_2) = Q_4(a_2) + U_4(a_2) = 1.794$

Pulling the argmax: 0

Bias of bandit 0: 0.5
Random noise: 0.8
Return: 1.0

$Q_4(a_0) = 0.8 + \frac{1}{2}(1.0 - 0.8) = 0.9$

Iteration 5:

Q-values:
$Q_5(a_0) = 0.9$
$U_5(a_0) = \sqrt{\frac{2\log 6}{3}} = 1.093$
$\text{UCB}_5(a_0) = Q_5(a_0) + U_5(a_0) = 1.993$

$Q_5(a_1) = 0.7$
$U_5(a_1) = \sqrt{\frac{2\log 6}{3}} = 1.093$
$\text{UCB}_5(a_1) = Q_5(a_1) + U_5(a_1) = 1.793$

$Q_5(a_2) = 0.525$
$U_5(a_2) = \sqrt{\frac{2\log 6}{2}} = 1.339$
$\text{UCB}_5(a_2) = Q_5(a_2) + U_5(a_2) = 1.864$

Pulling the argmax: 0

Bias of bandit 0: 0.5
Random noise: 0.5
Return: 1.0

$Q_5(a_0) = 0.9 + \frac{1}{3}(1.0 - 0.9) = 0.933$

Iteration 6:

Q-values:
$Q_6(a_0) = 0.933$
$U_6(a_0) = \sqrt{\frac{2\log 7}{4}} = 0.986$
$\text{UCB}_6(a_0) = Q_6(a_0) + U_6(a_0) = 1.919$

$Q_6(a_1) = 0.7$

$U_6(a_1) = \sqrt{\frac{2\log 7}{3}} = 1.139$
$\text{UCB}_6(a_1) = Q_6(a_1) + U_6(a_1) = 1.839$

$Q_6(a_2) = 0.525$
$U_6(a_2) = \sqrt{\frac{2\log 7}{2}} = 1.395$
$\text{UCB}_6(a_2) = Q_6(a_2) + U_6(a_2) = 1.92$

Pulling the argmax: 2

Bias of bandit 2: 0.125
Random noise: 0.6
Return: 0.725

$Q_6(a_2) = 0.525 + \frac{1}{2}(0.725 - 0.525) = 0.625$

Iteration 7:

Q-values:
$Q_7(a_0) = 0.933$
$U_7(a_0) = \sqrt{\frac{2\log 8}{4}} = 1.02$
$\text{UCB}_7(a_0) = Q_7(a_0) + U_7(a_0) = 1.953$

$Q_7(a_1) = 0.7$
$U_7(a_1) = \sqrt{\frac{2\log 8}{3}} = 1.177$
$\text{UCB}_7(a_1) = Q_7(a_1) + U_7(a_1) = 1.877$

$Q_7(a_2) = 0.625$
$U_7(a_2) = \sqrt{\frac{2\log 8}{3}} = 1.177$
$\text{UCB}_7(a_2) = Q_7(a_2) + U_7(a_2) = 1.802$

Pulling the argmax: 0

Bias of bandit 0: 0.5
Random noise: 0.1
Return: 0.6

$Q_7(a_0) = 0.933 + \frac{1}{4}(0.6 - 0.933) = 0.85$

Iteration 8:

Q-values:
$Q_8(a_0) = 0.85$
$U_8(a_0) = \sqrt{\frac{2 \log 9}{5}} = 0.937$
$\text{UCB}_8(a_0) = Q_8(a_0) + U_8(a_0) = 1.787$

$Q_8(a_1) = 0.7$
$U_8(a_1) = \sqrt{\frac{2 \log 9}{3}} = 1.21$
$\text{UCB}_8(a_1) = Q_8(a_1) + U_8(a_1) = 1.91$

$Q_8(a_2) = 0.625$
$U_8(a_2) = \sqrt{\frac{2 \log 9}{3}} = 1.21$
$\text{UCB}_8(a_2) = Q_8(a_2) + U_8(a_2) = 1.835$

Pulling the argmax: 1

Bias of bandit 1: 0.25
Random noise: 0.0
Return: 0.25

$Q_8(a_1) = 0.7 + \frac{1}{3}(0.25 - 0.7) = 0.55$

Iteration 9:

Q-values:
$Q_9(a_0) = 0.85$
$U_9(a_0) = \sqrt{\frac{2 \log 10}{5}} = 0.96$
$\text{UCB}_9(a_0) = Q_9(a_0) + U_9(a_0) = 1.81$

$Q_9(a_1) = 0.55$
$U_9(a_1) = \sqrt{\frac{2 \log 10}{4}} = 1.073$
$\text{UCB}_9(a_1) = Q_9(a_1) + U_9(a_1) = 1.623$

$Q_9(a_2) = 0.625$
$U_9(a_2) = \sqrt{\frac{2 \log 10}{3}} = 1.239$
$\text{UCB}_9(a_2) = Q_9(a_2) + U_9(a_2) = 1.864$

Pulling the argmax: 2

Bias of bandit 2: 0.125

Random noise: 0.9
Return: 1.0

$Q_9(a_2) = 0.625 + \frac{1}{3}(1.0 - 0.625) = 0.75$

2 Exploration and Real Physical Systems

Imagine you want to apply the algorithms from this lecture on a real physical system and some actions in some states may break your robot, so you have avoid them (but you do not know those states beforehand). However, the presented algorithms **need** to explore in order to find a good solution. Which exploration strategies from the lecture lead to problems and why? How would you approach exploration?

Solution

- Random/epsilon-greedy/decaying epsilon-greedy exploration can lead to failure states due to randomness (we might lose control when random actions are picked).
- *Optimism in the Face of Uncertainty* also has a problem. If we are uncertain about some part of the state-action space we are also uncertain about getting in a failure state.
- The same holds for *Optimistic Initialization*.

Safe Exploration is an active field of RL-research and is still an open question. – therefore there is no clear solution to this problem. However, we would like to suggest two approaches.

- **A model-based approach.** We can try to learn a dynamics model of the environment and check whether an action leads to a failure state. Here we make the assumption that we know the failure states, but not how to reach them.
- **We can also be cautious.** Assume that we start in a safe state and that small actions are more likely to be safe than big actions. Then we can employ ϵ-greedy exploration on the *small* subset of actions (We can do the same with an *Optimism in the Face of Uncertainty*-approach: choose uncertain actions, but only to some extent.). Over time, we can slowly grow the bubble of safe state-action pairs.