Online and Reinforcement Learning (2025)
Home Assignment 6

Jan Ljubas, pmj557

Contents

1 PPO

1.1 Return expressed as advantage over another policy

To prove the equation $J(\pi) = J(\pi_{ref}) + \mathbb{E}_{\mathcal{S}_0,\pi}\left[\sum_{t=0}^{\infty} \gamma^t A^{\pi_{ref}}(s_t, a_t) \mid s_0, \pi\right]$, we'll work through intermediate steps:

1. Start with the definition of the expected return:

$$J(\pi) = \mathbb{E}_{\mathcal{S}_0,\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$$

2. The advantage function is defined as:

$$A^{\pi_{ref}}(s_t, a_t) = Q^{\pi_{ref}}(s_t, a_t) - V^{\pi_{ref}}(s_t)$$

3. This means:

$$Q^{\pi_{ref}}(s_t, a_t) = A^{\pi_{ref}}(s_t, a_t) + V^{\pi_{ref}}(s_t)$$

4. The Q-function represents expected return starting from state s_t, taking action a_t, and following π_{ref} thereafter:

$$Q^{\pi_{ref}}(s_t, a_t) = \mathbb{E}\left[r(s_t, a_t) + \gamma V^{\pi_{ref}}(s_{t+1})\right]$$

5. For any policy π, we can express the return as:

$$J(\pi) = \mathbb{E}_{\mathcal{S}_0,\pi}\left[V^{\pi_{ref}}(s_0) + \sum_{t=0}^{\infty} \gamma^t (Q^{\pi_{ref}}(s_t, a_t) - V^{\pi_{ref}}(s_t))\right]$$

6. Since $V^{\pi_{ref}}(s_0)$ is independent of π (given fixed s_0):

$$\mathbb{E}_{\mathcal{S}_0,\pi}\left[V^{\pi_{ref}}(s_0)\right] = J(\pi_{ref})$$

7. Substituting the advantage function:

$$J(\pi) = J(\pi_{ref}) + \mathbb{E}_{\mathcal{S}_0,\pi}\left[\sum_{t=0}^{\infty} \gamma^t A^{\pi_{ref}}(s_t, a_t)\right]$$

8. This gives us:

$$J(\pi) = J(\pi_{ref}) + \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t A^{\pi_{ref}}(s_t, a_t) \mid s_0, \pi\right]$$

■

1.2 Clipping

To formalize this second condition (assuming $\frac{\pi(a_t,s_t)}{\pi_{ref}(a_t,s_t)} \notin [1-\epsilon, 1+\epsilon]$) :
When $\frac{\pi(a_t,s_t)}{\pi_{ref}(a_t,s_t)} > 1+\epsilon$ and $A^{\pi_{ref}}(s_t,a_t) > 0$:
• The clipping term becomes $\text{clip}\Big(\frac{\pi(a_t,s_t)}{\pi_{ref}(a_t,s_t)}, 1-\epsilon, 1+\epsilon\Big) = 1+\epsilon$
• The gradient would normally push toward increasing $\pi(s_t,a_t)$ because $A^{\pi_{ref}}(s_t,a_t) > 0$, but since we're already above $1+\epsilon$, we don't want to increase further.
• The gradient of the clipping term becomes zero with respect to $\pi(s_t,a_t)$
• "Not pointing away from the interval" means the gradient would push $\pi(s_t,a_t)$ back toward $[1-\epsilon, 1+\epsilon]$

When $\frac{\pi(a_t,s_t)}{\pi_{ref}(a_t,s_t)} < 1-\epsilon$ and $A^{\pi_{ref}}(s_t,a_t) < 0$:
• The clipping term becomes $\text{clip}\Big(\frac{\pi(a_t,s_t)}{\pi_{ref}(a_t,s_t)}, 1-\epsilon, 1+\epsilon\Big) = 1-\epsilon$
• The gradient would normally push toward decreasing $\pi(s_t,a_t)$ because $A^{\pi_{ref}}(s_t,a_t) < 0$, but since we're already below $1-\epsilon$, we don't want to decrease further.
• The gradient of the clipping term becomes zero with respect to $\pi(s_t,a_t)$

Formally, the condition "gradient direction does not point away from the interval" can be expressed as:

For $\frac{\pi(a_t,s_t)}{\pi_{ref}(a_t,s_t)} > 1+\epsilon$:
• The update is performed if $A^{\pi_{ref}}(s_t,a_t) \leq 0$ (which means the gradient points toward decreasing $\pi(s_t,a_t)$, pushing it back toward the interval)

For $\frac{\pi(a_t,s_t)}{\pi_{ref}(a_t,s_t)} < 1-\epsilon$:
• The update is performed if $A^{\pi_{ref}}(s_t,a_t) \geq 0$ (which means the gradient points toward increasing $\pi(s_t,a_t)$, pushing it back toward the interval).

1.3 Pi prime in PPO

The ratio $\frac{\pi(a_t,s_t)}{\pi_{ref}(a_t,s_t)}$ isn't always one because $\pi_{ref}(a_t,s_t)$ is the action probability under the old policy π_{ref} (parameters θ'), while π uses updated parameters θ. During training, θ changes, making $\pi \neq \pi_{ref}$.
Formally:

• $\pi_t = \pi(s_t,a_t)$ represents the probability of taking action a_t in state s_t under the policy that was used to collect the experience (the "old" policy with parameters θ').

• $\pi(a_t,s_t)$ represents the probability of taking the same action in the same state, but under the current policy being optimized (with updated parameters).

• During the PPO update process, the policy parameters are changing, which means the probabilities assigned to actions in given states will change too.

• If the policy hasn't been updated yet, or if the updates haven't affected the probabilities for these specific state-action pairs, then $\frac{\pi(a_t,s_t)}{\pi_t} = 1$. But after updates, this ratio can be either greater than or less than 1.

• This ratio is precisely what's used in importance sampling to correct for the fact that we're evaluating a different policy than the one used to collect the experiences.

• The clipping mechanism in PPO specifically constrains how far this ratio can deviate from 1, preventing too large policy updates that could destabilize learning.

So, the expression is not always one because the policy generating the experience (π with parameters θ') and the policy being updated can assign different probabilities to the same state-action pairs as the learning progresses.

2 Offline evaluation of the Bandit algorithms

2.1 UCB1

In the offline evaluation setting with importance sampling, and under the assumption of uniform sampling of the action a_{log}, it holds that $\mathbb{P}[a_{log}] = \frac{1}{K}$, where K is the number of actions. We further define the **importance-weighted reward**, the unbiased estimates of true rewards $r_t(a)$ defined for all actions:

$$\tilde{r_t}(a) = \frac{r_t(a)}{\mathbb{P}[a_{log}]} = Kr_t(a)$$

Now, this affects the UCB algorithm in terms of the value of the computed upper confidence bound, and in the analysis of the regret bound. Here's how:
The new upper confidence bound term is calculated as

$$U_t^{CB}(a) = \hat{\mu_{t-1}}(a) + \sqrt{\frac{K^2 \ln(t)}{N_{t-1}(a)}}$$

$$U_t^{CB}(a) = \frac{1}{N_{t-1}(a)} \sum_{t:A_t=a} Kr_t(a)\mathbb{I}[a_{log} = a_t] + \sqrt{\frac{K^2 \ln(t)}{N_{t-1}(a)}}$$

where the K^2 term appears inside the square root because of the fact that the empirical variance scales quadratically with the range (no "bandit magic"). I.e. bigger range (from [0,1] to [0, K]) caused the variance increase of the importance-weighted rewards $\tilde{r}_t$.

Here's the pseudocode, where the $U_t^{CB}(a)$ for each timestep t is defined as above and the indicator function $\mathbb{I}[a_{log} = a_t]$ is used to ensure that only logged actions contribute to the reward estimates:

```
N(a) = 0, cumulative_sum(a) = 0 for all a

For t <= K: Select each action once

For t > K:
    Select A_t = arg max_a UCB_t(a)
    Observe reward r_t(a_t)
    Compute R_t = K r_t(a_t) × I[a_log == a_t]  % importance-weighted reward
    Update:
        N(a_t) += I[a_log == a_t]       % can be 0 or 1
        cumulative_sum(a_t) += R_t      % can be 0 or K r_t(a_t)
```

2.1.1 Regret bound

Instead of repeating the whole proof as given in the slides for the regular UCB/LCB algorithms, it's just important to notice that the change with the importance-weighted rewards influences the regret bound in both "cases" we introduced: when $U_t^{CB}(a*) \leq \mu(a*)$ and when $U_t^{CB}(a) \geq \mu(a*)$. This further means that $N_T(a) \leq \frac{K^2 \ln(T)}{\Delta(a)^2}$, instead of just $\frac{\ln(T)}{\Delta(a)^2}$.

So, $\mathbb{E}[R_T(a)] = \sum_a \Delta(a)\mathbb{E}[N_T(a)] = O(\sum_a \frac{K^2 \ln T}{\Delta(a)})$, which is worse than the original $O(\sum_a \frac{\ln(T)}{\Delta(a)})$

2.1.2 Conclusion - no small variance exploitation

So in the case of UCB1, the exploration term is derived using the concentration bounds which depend on the variance of the reward estimates. Importance weighing increases the variance by a factor of K^2 and the UCB1 relies on the UCB based on the worst-case variance. As a result, it cannot take advantage of the cases where the variance is lower than the worst case.
In other words, UCB1's regret analysis relies on the reward range, not variance. The confidence term scales with K, leading to a range-dependent bound regardless of variance.

2.2 EXP3

Similarly to what I explained above, we are now working with $\mathbb{P}[a_{log}] = \frac{1}{K}$ and the **importance-weighted loss**, defined as

$$\tilde{l_t}(a) = \frac{l_t(a)}{\mathbb{P}[a_{log}]} = Kl_t(a) = K(1 - r_t(a))$$

Now, EXP3 uses probability distributions over actions, updated based on estimated losses. The standard EXP3 update is the same:

$$w_{t+1}(a) = w_t(a)e^{(-\eta \tilde{l_t}(a))}$$

2.2.1 Pseudoalgorithm

```
Initialize w(a) = 1 for all actions a
For each round t:
    Compute probabilities: p(a) = w(a) / sum(w)
    Sample action A_t from distribution p(a)
    Observe loss l_t(A_t)
    Compute: L_t = K * l_t(a_t) × I[a_log == A_t]
    Update (all) weights: w(a_t+1) = w(a_t) * exp(-eta * L_t)
```

2.2.2 Analysis

Since the weighted variance of the importance-weighted estimates is the same as in the original EXP3 algorithm, $\mathbb{E}[R_T]$ upper bound is asymptotically the same: $O(\sqrt{KT \ln K})$. The reason is exactly the so called "bandit magic", where the second moment (which is approximately the same as the variance in this case) is no larger than K.
We start with:

$$\sum_{t=1}^{T} \mathbb{E}\left[\sum_{a=1}^{K} p_t(a) l_{t,a}\right] - \min_a L_T(a) \leq \frac{\ln K}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \mathbb{E}\left[\sum_{a=1}^{K} p_t(a) (\tilde{l}_{t,a})^2\right]$$

$$\mathbb{E}[R_T(a)] \leq \frac{\ln K}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \sum_{a=1}^{K} p_t(a) \mathbb{E}\left[(K l_{t,a} \mathbb{I}[a_{log} = a_t])^2\right]$$

$$\mathbb{E}[R_T(a)] \leq \frac{\ln K}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \sum_{a=1}^{K} p_t(a) \mathbb{E}\left[(l_{t,a} K \frac{1}{K})^2\right]$$

$$\mathbb{E}[R_T(a)] \leq \frac{\ln K}{\eta} + \frac{\eta}{2} KT$$

Since this part is the same as with the original EXP3, the following is as well. After optimizing this w.r.t. η, we get $\eta* = \sqrt{\frac{2 \ln K}{KT}}$, which yields the final regret bound: $\mathbb{E}[R_T(a)] = O(\sqrt{KT \ln K})$.

So, the importance sampling introduces a factor of K in the loss estimates. However, the variance remains at order K rather than K^2 due to uniform sampling by the logging policy and the fact that $\mathbb{E}[\mathbb{I}[a_{log} = a_t])^2] = p_{log}(a) = \frac{1}{K}$. This, finally, allows the regret bound to remain the same as before!

2.3 Anytime EXP3

Well, since the text said only the new learning rate and the new regret bound were required for solving the task, here they are:

$$\eta_t = \sqrt{\frac{\ln K}{2Kt}}$$

$$\mathbb{E}[R_t] = O(\sqrt{2KT \ln K}) \text{ for all t} \leq \text{T}$$