OReL Assignment 7

bkx591

April 2024

1 PPO (25 points)

1.1 Return expressed as advantage over another policy (5 points)

The expected return of a policy π' can be expressed in terms of its advantage over another policy π and $J(\pi)$:

$$J(\pi') = J(\pi) + \mathbb{E}\left\{ \sum_{t=1}^{\infty} \gamma^t A^{\pi}(s_t, a_t) \middle| s_0, \pi' \right\}$$

The advantage A quantifies the advantage of doing a in state s (and following π afterwards) instead of following π:

$$A^{\pi}(s, a) = Q^{\pi}(s, a) - V^{\pi}(s)$$

where $Q^{\pi}(s, a)$ is the expected return of taking action a in state s and thereafter following policy π, and $V^{\pi}(s)$ is the expected return of following policy π from state s. Now, if we consider another policy π' and we want to express its expected return in terms of its advantage over the original policy π, we recall the definitions of the value function

$$V^{\pi}(s) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s]$$

and the action value (Q) function

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

Now, the expected (discounted) return for a policy π starting in state s_0 is given by:

$$J(\pi) = \mathbb{E}_{\pi}\left\{ \sum_{t=0}^{\infty} \gamma^t r_t \middle| s_0, \pi \right\}$$

and we can expand $J(\pi')$ by adding the Q^π and subtracting V^π, because $J(\pi)$ is the expected return under π, and we're adding the difference when actions are chosen under π' s.t.

$$J(\pi') = J(\pi) + \mathbb{E}\left\{\sum_{t=1}^{\infty} \gamma^{t-1}(Q^\pi(s_t, a_t) - V^\pi(s_t))\right\}$$

and by substituting $Q^\pi(s_t, a_t)$ with $A^\pi(s_t, a_t) + V^\pi(s_t)$ and noting that $V^\pi(s_t)$ is the expected return starting from s_t and following π, the expected return for π' becomes:

$$J(\pi') = J(\pi) + \mathbb{E}_{\pi'}\left\{\sum_{t=1}^{\infty} \gamma^{t-1} A^\pi(s_t, a_t)\right\}$$

1.2 Clipping (10 points)

In the clipping component of the Proximal Policy Optimization (PPO) algorithm, the objective is designed to prevent the policy from changing too much, which is achieved by constraining/clipping the ratio of the new policy probability to the old policy probability. This ratio is noted as:

$$\frac{\pi'(a_t, s_t)}{\pi(a_t, s_t)}$$

Where $\pi'(a_t, s_t)$ is the probability under the new policy of taking action a_t in state s_t, and $\pi(a_t, s_t)$ is the probability under the old policy. The PPO-Clip objective uses a clipped ratio to modify the advantage function $\hat{A}^\pi(s_t, a_t)$. The clipped ratio is defined as:

$$\text{clip}\left(\frac{\pi'(a_t, s_t)}{\pi(a_t, s_t)}, 1-\epsilon, 1+\epsilon\right)$$

The statement about the gradient-based update getting closer to this range refers to ensuring the updated policy doesn't move the probability ratio outside the range $[1-\epsilon, 1+\epsilon]$. In other words, the clipping operation restricts the updated policy's probability ratio, so it doesn't become too large or too small compared to the old policy's probability ratio.

We are asked to formalize and prove the condition when the gradient direction does not point away from the interval (slide 44/51) $[1-\epsilon, 1+\epsilon]$, assuming the first condition $\frac{\pi'(a_t,s_t)}{\pi(a_t,s_t)} \notin [1-\epsilon, 1+\epsilon]$ is not met. To formalize this condition, consider the following scenarios for the ratio $r_t(\theta) = \frac{\pi'(a_t,s_t)}{\pi(a_t,s_t)}$ in which $r_t(\theta) \notin [1-\epsilon, 1+\epsilon]$:

1. $r_t(\theta) > 1+\epsilon$: In this case, the update should not increase $r_t(\theta)$ further. Therefore, the update is only applied if it brings $r_t(\theta)$ closer to $1+\epsilon$. This happens when the advantage function $\hat{A}^\pi(s_t, a_t)$ is negative (since we would want to reduce the action's probability), and thus, the gradient should be negative to reduce $r_t(\theta)$. ($\nabla_\theta J(\theta)$ should be negative.)

2. $r_t(\theta) < 1 - \epsilon$: Here, the update should not decrease $r_t(\theta)$ further. Therefore, the update is only applied if it brings $r_t(\theta)$ closer to $1 - \epsilon$. This occurs when the advantage function $\hat{A}^{\pi}(s_t, a_t)$ is positive (since we would want to increase the action's probability), and thus, the gradient should be positive to increase $r_t(\theta)$ ($\nabla_\theta J(\theta)$ should be positive).

The proofs for formalization comes from the behaviour of clipping mechanism under the given settings.

1.3 Pi prime in PPO (10 points)

In PPO, we're basically teaching the policy π' with parameters θ' to pick actions in different states during the "Gather experience" stage (slide 46/51). The chances of these actions happening, given the policyπ', are recorded as p_t. Now, when we tweak the policy using PPO, it starts to give different weights to actions because it's getting better at figuring out what works best, based on what it's seen so far.

PPO keeps track of how much it's changing the policy using the ratio $\frac{\pi'(a_t, s_t)}{p_t}$. This ratio tells us how much the policy has learned since we last checked. It's not always going to be one because that would mean we haven't learned anything new and the policy hasn't changed, which is not the what we want.

This ratio bouncing around from one shows us that the policy is getting updated with every learning round. Its to make sure that we're not just blindly following old policies but actually incorporating new learnings.

2 MBIE and Prior Knowledge on Support Sets (25 points)

(i) The precise mathematical form of the revised confidence sets $C_{s,a}$ incorporating the idea of a support set, $\mathcal{K}_{s,a}$ of the distribution $P(\cdot|s, a)$ is:

$$C_{s,a} = \{p \in \Delta(S) \mid \forall s' \notin \mathcal{K}_{s,a}, p(s'|s, a) = 0\}$$

(ii) The support is already implemented in the riverswim class, so to adapt the solution of MBIE, I simply pass the support as argument to the max_proba() function, with this minor modification:

```
...
# Apply confidence interval adjustments only within the support set.
for ss in support[s][a]:
    max_p[ss] = self.hatP[s, a][ss] + (self.confP[s, a] / 2)
```

Resulting in an algorithm that makes massively fewer ϵ-bad steps as seen in Figure 1.

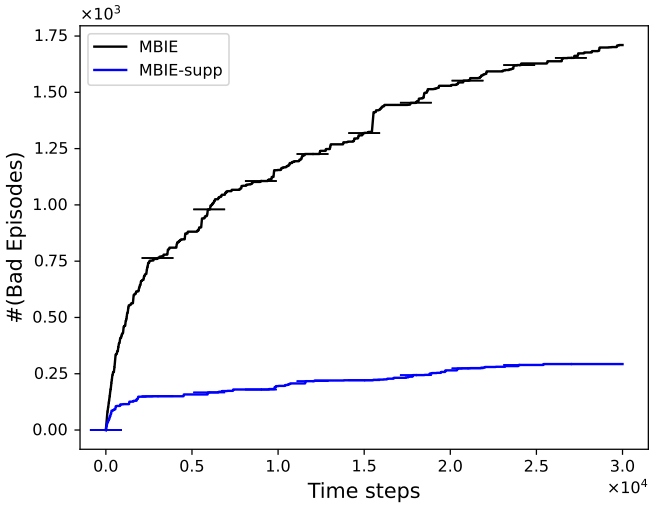

Figure 1: MBIE with and without support

3 Offline Evaluation of Bandit Algorithms (50 points)

The straightforward way of evaluating algorithms for online learning with limited feedback in real life is undesirable for many reasons.

- **Time to Convergence**: One such reason is simply that the time it takes for the algorithm to develop a proper strategy, might be disproportionate to what is reasonable. By the time such a favorable algorithm is explored, many unfavorable or sub-optimal decisions will have been made.
- **Risk of Sub-optimal decisions**: As mentioned above, many sub-optimal decisions will have been made. But even more undesirable consequences might form from the exploration component of online learning algorithms. The property of trying out less-favored actions to gain information, could lead to undesirable outcomes for many reasons - especially in real world scenarios such as medical treatments of financial investments. In simulated environments, these risks can be mitigated.

The Theoretical Part of the Task

(a) Modify UCB

The UCB1 algorithm with improved parametrization is designed for $r_{t,a} \in [0, 1]$ as is defined as follows

$$U_t^{CB}(a) = \hat{\mu}_{t-1}(a) - \sqrt{\frac{\ln t}{N_{t-1}(a)}}$$

as mentioned in the assignment, we need to scale the reward to $[0, K]$, specifically, we need to scale the confidence interval s.t.

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

Recall Hoeffding's Inequality from Theorem 2.3 with a fixed confidence in mind:

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} X_i - \mu \geq \sqrt{\frac{\ln \frac{1}{\delta}}{2n}}\right) \leq \delta$$

When we analyze Hoeffding's Inequality with r.t. the r.v. bounded in $[0, K]$ we get

$$\mathbb{P}\left(\sum_{i=1}^{n} X_i - \mu \geq \varepsilon\right) \leq e^{-2\varepsilon^2 n^2 / \sum_{i=1}^{n} K^2} = e^{-2\varepsilon^2 n / K^2}$$

This gives us a $\delta = e^{-2\varepsilon^2 n/K^2}$ and an $\varepsilon = K\sqrt{\frac{\ln \frac{1}{\delta}}{2n}}$. We then translate this scaling to the modified UCB1. Next up, we want to bound the regret, which is why we want to bound the expected number of times $U_t^{CB}(a) \leq U_t^{CB}(a^*)$. This is bounded by:

1. The time it takes for confidence intervals to start working (The expected number of times $U_t^{CB}(a) \leq \mu(a^*)$):
   If $\hat{\mu}_t(a) > \mu(a) - K\sqrt{\frac{\ln t}{N_{t-1}(a)}}$ then

$$U_t^{CB}(a) > \mu(a) - 2K\sqrt{\frac{\ln t}{N_{t-1}(a)}} \tag{1}$$

$$= \mu(a^*) + \Delta(a) - 2K\sqrt{\frac{\ln t}{N_{t-1}(a)}} \tag{2}$$

$$\tag{3}$$

   And so we have $U_t^{CB}(a) \leq \mu(a^*)$ if $\Delta(a) < 2K\sqrt{\frac{\ln t}{N_{t-1}(a)}}$ s.t.

$$N_t(a) \leq \frac{4K^2 \ln T}{\Delta(a)^2} \tag{4}$$

2. the expected number of times the confidence intervals fail ($U_t^{CB}(a^*) \leq \mu(a^*)$):

$$\mathbb{P}\left(U_t^{CB}(a^*) \leq \mu(a^*)\right) = \mathbb{P}\left(\hat{\mu}_{t-1}(a) - K\sqrt{\frac{\ln t}{N_{t-1}(a)}} \geq \mu(a^*)\right) \tag{5}$$

$$= \mathbb{P}\left(\hat{\mu}_{t-1}(a) - \mu(a^*) \geq K\sqrt{\frac{\ln t}{N_{t-1}(a)}}\right) \tag{6}$$

Now we introduce $X_1, ...X_T$ r.v. with the same distribution as ℓ_{t,a^*}, which in this case is $X_i \in [0, K]$, and let $\bar{\mu}_s = \frac{1}{s}\sum_{i=1}^{s} X_i$

$$\mathbb{P}(...) \leq \mathbb{P}\left(\exists s : \bar{\mu}_s - \mu(a^*) \geq K\sqrt{\frac{\ln t}{s}}\right) \tag{7}$$

$$= \mathbb{P}\left(\exists s : s\bar{\mu}_s - s\mu(a^*) \geq sK\sqrt{\frac{\ln t}{s}}\right) \tag{8}$$

$$= \mathbb{P}\left(\exists s : \sum_{i=1}^{s} X_i - \mathbb{E}\left[\sum_{i=1}^{s} X_i\right] \geq sK\sqrt{\frac{\ln t}{s}}\right) \tag{9}$$

And by union bound

$$\mathbb{P}(...) \leq \sum_{s=1}^{t} \mathbb{P}\left(\sum_{i=1}^{s} X_i - \mathbb{E}\left[\sum_{i=1}^{s} X_i\right] \geq sK\sqrt{\frac{\ln t}{s}}\right) \tag{10}$$

And by Hoeffding

$$\mathbb{P}(...) \leq \sum_{s=1}^{t} e^{-2(sK\sqrt{\frac{\ln t}{s}})^2 / \sum_{i=1}^{t} K^2} \tag{11}$$

$$= \sum_{s=1}^{t} e^{-2s^2K^2\frac{\ln t}{s}/sK^2} \tag{12}$$

$$= \sum_{s=1}^{t} e^{-2\ln t} \tag{13}$$

$$= \sum_{s=1}^{t} \frac{1}{t^2} \tag{14}$$

$$= \frac{1}{t} \tag{15}$$

s.t. $\mathbb{E}[F(a^*)] = \sum_{t=1}^{T} \frac{1}{t} \leq \ln T + 1$

Combining above derivations we have that

$$\mathbb{E}\left[N_T(a)\right] \leq \frac{4K^2 \ln T}{\Delta(a)^2} + \mathbb{E}\left[F(a^*)\right] + \mathbb{E}\left[F(a)\right] \tag{16}$$

$$\leq \frac{4K^2 \ln T}{\Delta(a)^2} + 2(\ln T + 1) \tag{17}$$

The total expected regret is then:

$$\bar{R}_t(a) = \sum_a \Delta(a)\mathbb{E}\left[N_T(a)\right] \tag{18}$$

$$\leq \sum_a \Delta(a) \left(\frac{4K^2 \ln T}{\Delta(a)^2} + 2(\ln T + 1) \right) \tag{19}$$

- Modified UCB Algorithm
  - Play each arm once
  - For $t = K+1, K+2, ...$
    - Play $A_t = \arg\min_a U_t^{CB}(a)$

(b) Explain unexploitable small variance

In the modified UCB1, the small variance cannot be exploited due to the static nature of the upper confidence bound. Unlike EXP3, where the probability of selecting an action is affected by the observed variance, the UCB1's confidence bound remain static and does not tighten with lower variance.

(c) Modify EXP3

When modifying EXP3 to work with importance-weighted losses, we must address the increased variance in the estimated losses. The importance-weighted loss must be re-scaled to the range $[0, 1]$ using the importance weight derived from the uniform logging policy's probability of action selection, and we must tune the loss to when the played action a_t at time t matches the logged action a_t^{log}, so let's address the observed loss we suffer:

$$l_{t,a}^{IS} = \frac{l_{t,a}\mathbb{1}[A_t = a_t]\mathbb{1}[A_t = a_t^{log}]}{\pi(a_t)}$$

$$= \frac{l_{t,a}\mathbb{1}[A_t = a_t]\mathbb{1}[A_t = a_t^{log}]}{\frac{1}{K}}$$

$$= K l_{t,a}\mathbb{1}[A_t = a_t]\mathbb{1}[A_t = a_t^{log}]$$

and then look to the importance-weighted loss estimator:

$$\begin{aligned}
\tilde{\ell}_{t,a} &= \frac{Kl_{t,a}\mathbb{1}[A_t = a_t]\mathbb{1}[A_t = a_t^{log}]}{p_t(a)} \\
\mathbb{E}\left[\tilde{\ell}_{t,a}\right] &= \mathbb{E}\left[\frac{Kl_{t,a}\mathbb{1}[A_t = a_t]\mathbb{1}[A_t = a_t^{log}]}{p_t(a)}\right] \\
&= Kl_{t,a}\mathbb{E}\left[\frac{\mathbb{1}[A_t = a_t]}{p_t(a)}\right]\mathbb{E}\left[\mathbb{1}[A_t = a_t^{log}\right] \\
&= Kl_{t,a}\mathbb{E}\left[\frac{\mathbb{1}[A_t = a_t]}{p_t(a)}\right]\frac{1}{K} \\
&= Kl_{t,a}\mathbb{E}_{A_1,...,A_{t-1}}\left[\mathbb{E}_{A_t}\left[\frac{\mathbb{1}[A_t = a_t]}{p_t(a)} \mid A_1, ..., A_{t-1}\right]\right]\frac{1}{K} \\
&= l_{t,a}\mathbb{E}_{A_1,...,A_{t-1}}\left[\mathbb{E}_{A_t}\left[\frac{\mathbb{1}[A_t = a_t]}{p_t(a)} \mid A_1, ..., A_{t-1}\right]\right] \\
&= l_{t,a}
\end{aligned}$$

Pseudocode for Modified EXP3:

- $\forall a : L_0(a) = 0$
- For $t = 1, 2, ...$
  - $\forall a : p_t(a) = \frac{e^{-\eta_t L_{t-1}(a)}}{\sum_{a'} e^{-\eta_t L_{t-1}(a')}}$
  - $A_t \sim p_t$ (A_t is independent of the logged action)
  - [Observe ℓ^{IS}_{t,A_t}]
  - $\forall a : L_t(a) = L_{t-1}(a) + \frac{Kl_{t,a}\mathbb{1}[A_t=a_t]\mathbb{1}[A_t=a_t^{log}]}{p_t(a)}$ (modified observed loss)

(d) Anytime modified EXP3

The new learning rate will look like this

$$\eta_t = \frac{\eta_1}{(\sqrt{2})^{t-1}}$$

Recall the regret bound on EXP3:

$$\mathbb{E}\left[R_T\right] \leq \sqrt{2KT\ln K}$$

and the new regret bound is larger by a factor of $\sqrt{2}$ and holds for all t and not just for one specific time T s.t.

$$\mathbb{E}\left[R_t\right] \leq \sqrt{4Kt\ln K}$$