Homework 2

6501 Reinforcement Learning (Fall 2025)

Submission deadline: 11:59pm, October 14

The latex template is here. We do not provide new starter code for Problem 3 — please use the same file from Homework 1. Submit a .pdf and a .py file to Gradescope.

1 KL-Regularized Policy Improvement = Exponential Weight Update

In this problem, we show the equivalence between KL-regularized policy improvement and exponential weight updates (Page 7 of this slide). Let Δ_A denote the A-dimensional probability simplex:

$$\Delta_A = \left\{ \pi \in \mathbb{R}^A : \sum_{a=1}^{A} \pi(a) = 1 \text{ and } \forall a, \pi(a) \geq 0 \right\}.$$

The KL-regularized policy update algorithm is defined as

$$\pi_{t+1} = \operatorname*{argmax}_{\pi \in \Delta_A} \left\{ \langle \pi, r_t \rangle - \frac{1}{\eta} \mathrm{KL}(\pi, \pi_t) \right\} = \operatorname*{argmax}_{\pi \in \Delta_A} \left\{ \sum_{a=1}^{A} \pi(a) r_t(a) - \frac{1}{\eta} \sum_{a=1}^{A} \pi(a) \log \frac{\pi(a)}{\pi_t(a)} \right\}. \tag{1}$$

(a) (5%) Apply the Lagrange multiplier theorem (Theorem 1) on the maximization problem Eq. (1). Write the first-order optimality condition (Eq. (11)) that must hold for π_{t+1} and the Lagrange multiplier λ. You may assume $\pi_t(a) \neq 0$ and $\pi_{t+1}(a) \neq 0$ for all a. You do not need to justify the conditions of Theorem 1.
Hint: Take $g(\pi) = \sum_a \pi(a) - 1$ as the equality constraint and $\Omega = \{\pi \in \mathbb{R}^A : \forall a, \pi(a) \geq 0\}$ in Theorem 1.

Proof. Define

$$f(\pi) = \sum_{a=1}^{A} \pi(a)\, r_t(a) - \frac{1}{\eta} \sum_{a=1}^{A} \pi(a) \log \frac{\pi(a)}{\pi_t(a)}, \qquad g(\pi) = \sum_{a=1}^{A} \pi(a) - 1,$$

and the Lagrangian

$$\mathcal{L}(\pi, \lambda) = f(\pi) + \lambda\, g(\pi).$$

At an interior optimum (we assume $\pi_t(a) \neq 0$ and $\pi_{t+1}(a) \neq 0$ for all a), the first-order conditions are, for each a,

$$\frac{\partial \mathcal{L}}{\partial \pi(a)}\Big|_{\pi=\pi_{t+1}} = r_t(a) - \frac{1}{\eta}\left(\log \frac{\pi_{t+1}(a)}{\pi_t(a)} + 1 \right) + \lambda = 0.$$

Equivalently,

$$\log \frac{\pi_{t+1}(a)}{\pi_t(a)} = \eta\big(r_t(a) + \lambda\big) - 1, \qquad \forall a.$$

□

(b) (5%) From (a), prove that

$$\pi_{t+1}(a) = \frac{\pi_t(a) \exp(\eta r_t(a))}{\sum_{a'=1}^{A} \pi_t(a') \exp(\eta r_t(a'))}. \tag{2}$$

Proof. Exponentiating the result from 1a gives

$$\pi_{t+1}(a) = \pi_t(a)\exp(\eta r_t(a))\exp(\eta\lambda - 1)\,.$$

Let $C \coloneqq \exp(\eta\lambda - 1)$ (independent of a). Enforce normalization:

$$1 = \sum_{a=1}^{A}\pi_{t+1}(a) = C\sum_{a=1}^{A}\pi_t(a)\exp(\eta r_t(a)) \quad \Rightarrow \quad C = \frac{1}{\sum_{a'}\pi_t(a')\exp(\eta r_t(a'))}.$$

Plugging C back yields the claimed exponential-weights form:

$$\pi_{t+1}(a) = \frac{\pi_t(a)\exp(\eta r_t(a))}{\sum_{a'}\pi_t(a')\exp(\eta r_t(a'))}.$$

□

2 Regret Bound for Exponential Weight Update

In this problem, we will prove the regret bound for the exponential weight algorithm, i.e., Eq. (2). This will complete the proof for the theorem on Page 8 of this slide. We assume the initial policy π_1 is a uniform distribution over actions $\{1, 2, \ldots, A\}$. Let $a^\star \in \{1, 2, \ldots, A\}$ be any fixed action that we would like to compare the learner's performance with.

To facilitate the analysis, define

$$S_t(a) = \sum_{\tau=1}^{t} r_\tau(a) \qquad \text{and} \qquad \Phi_t = \frac{1}{\eta}\log\left(\sum_{a=1}^{A}\exp\left(\eta S_t(a)\right)\right). \tag{3}$$

(a) (5%) Argue that the update rule Eq. (2) can also be written as

$$\pi_{t+1}(a) = \frac{\exp(\eta S_t(a))}{\sum_{a'=1}^{A}\exp(\eta S_t(a'))}.$$

Proof. We are given the multiplicative-weights update

$$\pi_{t+1}(a) = \frac{\pi_t(a)\,e^{\eta r_t(a)}}{\sum_{a'}\pi_t(a')\,e^{\eta r_t(a')}}.$$

Ignoring the normalization (the denominator) for the moment, we can write this as

$$\begin{aligned}
\pi_{t+1}(a) &\propto \pi_t(a)\,e^{\eta r_t(a)} \\
&\propto \pi_{t-1}(a)\,e^{\eta r_{t-1}(a)}\,e^{\eta r_t(a)} \\
&\propto \cdots \\
&\propto \pi_1(a)\,e^{\left(\eta\sum_{\tau=1}^{t} r_\tau(a)\right)}. \\
&\propto \pi_1(a)\,e^{\eta S_t(a)}.
\end{aligned}$$

Now we normalize over actions. Divide by the sum over all actions a':

$$\pi_{t+1}(a) = \frac{\pi_1(a)\,e^{\eta S_t(a)}}{\sum_{a'}\pi_1(a')\,e^{\eta S_t(a')}}.$$

If the initial distribution π_1 is uniform over A actions, then $\pi_1(a) = \frac{1}{A}$ for all a, so it cancels itself out:

$$\pi_{t+1}(a) = \frac{e^{\eta S_t(a)}}{\sum_{a'} e^{\eta S_t(a')}}.$$

□

(b) (5%) Prove that

$$\Phi_t - \Phi_{t-1} = \frac{1}{\eta} \log \left(\sum_{a=1}^{A} \pi_t(a) \exp(\eta r_t(a)) \right). \tag{4}$$

Hint: By direct calculation using the definition of Φ_t and the fact $\pi_t(a) = \frac{\exp(\eta S_{t-1}(a))}{\sum_{a'=1}^{A} \exp(\eta S_{t-1}(a'))}$ which has been prove in (a).

Proof.

$$\Phi_t \;=\; \frac{1}{\eta} \log \sum_a \exp\big(\eta S_t(a)\big), \qquad S_t(a) \;=\; S_{t-1}(a) + r_t(a).$$

$$\begin{aligned}
\Phi_t - \Phi_{t-1} &= \frac{1}{\eta} \log \left(\frac{\sum_a \exp\big(\eta S_t(a)\big)}{\sum_a \exp\big(\eta S_{t-1}(a)\big)} \right) \\
&= \frac{1}{\eta} \log \left(\frac{\sum_a \exp\big(\eta S_{t-1}(a)\big)\, \exp\big(\eta r_t(a)\big)}{\sum_a \exp\big(\eta S_{t-1}(a)\big)} \right) \\
&= \frac{1}{\eta} \log \left(\sum_a \underbrace{\frac{\exp\big(\eta S_{t-1}(a)\big)}{\sum_{a'} \exp\big(\eta S_{t-1}(a')\big)}}_{=\,\pi_t(a)} \exp\big(\eta r_t(a)\big) \right) \\
&= \frac{1}{\eta} \log \left(\sum_a \pi_t(a)\, \exp\big(\eta r_t(a)\big) \right).
\end{aligned}$$

$\square$

(c) (5%) Continue from Eq. (4) with the inequalities $\exp(x) \le 1 + x + x^2$ for $x \le 1$ and $\log(1 + x) \le x$ for any x to show that

$$\Phi_t - \Phi_{t-1} \le \sum_{a=1}^{A} \pi_t(a) r_t(a) + \eta \sum_{a=1}^{A} \pi_t(a) r_t(a)^2$$

whenever $\eta r_t(a) \le 1$ for all a.
Hint: Upper bound the $\exp(\ldots)$ term in Eq. (4) with the first inequality. Then try to apply the second inequality.

Proof. From (b):

$$\Phi_t - \Phi_{t-1} = \frac{1}{\eta} \log\Big(\sum_a \pi_t(a)\, \exp(\eta r_t(a)) \Big)$$

Assume $\eta r_t(a) \le 1$ for all a. Using $\exp(x) \le 1 + x + x^2$ for $x \le 1$,

$$\sum_a \pi_t(a)\, \exp(\eta r_t(a)) \;\le\; \sum_a \pi_t(a)\, \big(1 + \eta r_t(a) + \eta^2 r_t(a)^2\big) = 1 + \eta \sum_a \pi_t(a) r_t(a) + \eta^2 \sum_a \pi_t(a) r_t(a)^2.$$

Define

$$x \;=\; \eta \sum_a \pi_t(a) r_t(a) + \eta^2 \sum_a \pi_t(a) r_t(a)^2,$$

so the far RHS in the equation before is $1 + x$. Taking the $\log$ of the far LHS and far RHS and dividing by η, we get

$$\Phi_t - \Phi_{t-1} \;=\; \frac{1}{\eta} \log\Big(\sum_a \pi_t(a)\, \exp(\eta r_t(a)) \Big) \;\le\; \frac{1}{\eta} \log(1 + x).$$

Finally, with $\log(1+u) \le u$ for $u > -1$ (which applies since $1 + x > 0$),

$$\Phi_t - \Phi_{t-1} \;\le\; \frac{1}{\eta}x \;=\; \sum_a \pi_t(a) r_t(a) \;+\; \eta \sum_a \pi_t(a) r_t(a)^2.$$

Thus, under $\eta\, r_t(a) \le 1$ for all a,

$$\boxed{\Phi_t - \Phi_{t-1} \;\le\; \sum_a \pi_t(a) r_t(a) \;+\; \eta \sum_a \pi_t(a) r_t(a)^2}$$

as required.

□

(d) (5%) Show that $\Phi_T \ge S_T(a^\star) = \sum_{t=1}^T r_t(a^\star)$ and $\Phi_0 = \frac{\log A}{\eta}$.
Hint: Use the definition of Φ_t in Eq. (3).

Proof.

$$\Phi_T \;=\; \frac{1}{\eta} \log \sum_a \exp\bigl(\eta S_T(a)\bigr).$$

Since $\sum_a \exp(\eta S_T(a)) \ge \exp(\eta S_T(a^\star))$,

$$\Phi_T \;\ge\; \frac{1}{\eta} \log \exp\bigl(\eta S_T(a^\star)\bigr) \;=\; S_T(a^\star) \;=\; \sum_{t=1}^T r_t(a^\star).$$

For $t = 0$, by definition $S_0(a) = 0$ (empty sum), hence

$$\Phi_0 \;=\; \frac{1}{\eta} \log \sum_a \exp(0) \;=\; \frac{1}{\eta} \log A.$$

□

(e) (5%) Combine (c) and (d) to show that

$$\sum_{t=1}^T r_t(a^\star) - \sum_{t=1}^T \sum_{a=1}^A \pi_t(a) r_t(a) \le \frac{\log A}{\eta} + \eta \sum_{t=1}^T \sum_{a=1}^A \pi_t(a) r_t(a)^2$$

if $\eta r_t(a) \le 1$ for all t and a.
Hint: $\Phi_T - \Phi_0 = \sum_{t=1}^T (\Phi_t - \Phi_{t-1})$. Upper bound the right-hand side using (c), and lower bound the left-hand side using (d).

Proof. From part (c), for each t (assuming $\eta\, r_t(a) \le 1$ for all a),

$$\Phi_t - \Phi_{t-1} \le \sum_a \pi_t(a)\, r_t(a) + \eta \sum_a \pi_t(a)\, r_t(a)^2.$$

Summing over $t = 1, \ldots, T$ and telescoping,

$$\Phi_T - \Phi_0 \le \sum_{t=1}^T \sum_a \pi_t(a)\, r_t(a) + \eta \sum_{t=1}^T \sum_a \pi_t(a)\, r_t(a)^2.$$

From part (d), $\Phi_T \ge \sum_{t=1}^T r_t(a^\star)$ and $\Phi_0 = \frac{1}{\eta} \log A$. Therefore,

$$\sum_{t=1}^T r_t(a^\star) - \sum_{t=1}^T \sum_a \pi_t(a)\, r_t(a) \le \frac{\log A}{\eta} + \eta \sum_{t=1}^T \sum_a \pi_t(a)\, r_t(a)^2.$$

□

3 Implementing Contextual Bandit Algorithms

We continue to implement contextual bandits algorithms in the same problem in Homework 1. We do not provide additional starter code — just add your code to that in Homework 1. Please follow the same instruction as in Homework 1 to present your results. In this problem, you will implement Proximal Policy Optimization (PPO) and Policy Gradient (PG). Please try to reach a best "overall score" of 0.6 for PPO, and 0.55 for PG (which will ensure full score on the implementation part).

You are free to change the default hyperparameters. In the tables below, you may also change the values of hyperparameters or add additional ones if you feel that the given values cannot reflect the trend.

3.1 PPO

Algorithm 1 Proximal Policy Optimization (without clipping)

Default hyperparameters: $N = 256$, $M = 30$, and $1/\eta = 0.1$.
Randomly initialize a policy network π_θ that takes contexts as input and outputs an action distribution.
Let θ_1 be the initial weights for the policy network.
If using adaptive baseline, additionally initialize a baseline network b_ϕ.
for $t = 1, \dots, T$ **do**
  for $n = 1, \dots, N$ **do**
    Receive context $x_{t,n}$.
    Sample action $a_{t,n} \sim \pi_{\theta_t}(\cdot|x_{t,n})$
    Receive reward $r_{t,n}$.
  $\theta \leftarrow \theta_t$
  for $m = 1, \dots, M$ **do**

$$\theta \leftarrow \theta + \lambda \nabla_\theta \left\{ \frac{1}{N} \sum_{n=1}^{N} \left[\frac{\pi_\theta(a_{t,n}|x_{t,n})}{\pi_{\theta_t}(a_{t,n}|x_{t,n})} (r_{t,n} - b_{t,n}) - \frac{1}{\eta} \left(\frac{\pi_\theta(a_{t,n}|x_{t,n})}{\pi_{\theta_t}(a_{t,n}|x_{t,n})} - 1 - \log \frac{\pi_\theta(a_{t,n}|x_{t,n})}{\pi_{\theta_t}(a_{t,n}|x_{t,n})} \right) \right] \right\}, \tag{5}$$

    where

$$b_{t,n} = \begin{cases} b & \text{for fixed baseline} \\ b_\phi(x_{t,n}) + b' & \text{for adaptive baseline} \end{cases} \tag{6}$$

    If using adaptive baseline, update

$$\phi \leftarrow \phi - \lambda' \nabla_\phi \left[\frac{1}{N} \sum_{n=1}^{N} \left(b_\phi(x_{t,n}) - r_{t,n} \right)^2 \right]. \tag{7}$$

  $\theta_{t+1} \leftarrow \theta$

An important implementation detail for Algorithm 1: In Eq. (5), the term $\pi_{\theta_t}(a_{t,n} \mid x_{t,n})$ in the denominator should be treated as a constant with respect to the gradient operator ∇_θ. In practice, this means applying the `detach()` function to the tensor, so that gradients are not propagated through it.

In general, one may perform *mini-batch* gradient descent in Eq. (5). That means in each iteration $m = 1, 2, \dots, M$, we only use B out of the N samples to perform the update for some $B < N$. This is the version we presented on Page 41 of this slide, and is also the more standard PPO for larger-scale problems. In this homework, we do it without mini-batching for simplicity.

3.1.1 The effect of Baseline

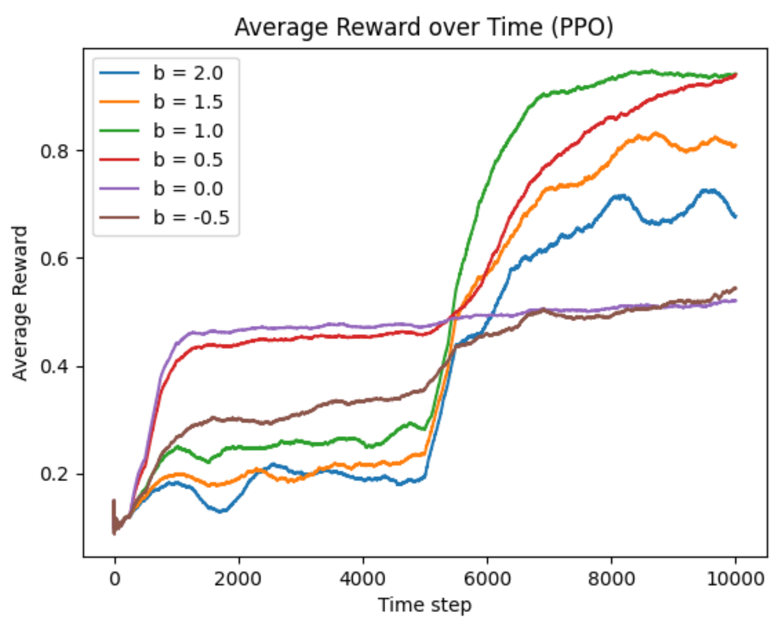
For this part, you could set $1/\eta = 0.1$ in Eq. (5) or any other fixed value. It will be tuned in Section 3.1.2.

Fixed Baseline

(a) (5%) Implement Algorithm 1 with fixed baseline (so Eq. (7) can be omitted and use the first option in Eq. (6)) and, for different values of b, record in the table below the average reward in Phase 1, Phase 2, and over the entire horizon.

b	Phase 1	Phase 2	Overall
2	0.1817	0.6228	0.4022
1.5	0.2001	0.7221	0.4611
1	0.2471	0.8636	0.5554
0.5	0.4226	0.7889	0.6057
0	0.4432	0.5046	0.4739
-0.5	0.2997	0.4937	0.3967

(b) (5%) Plot the running average reward over time for your experiments in (a) (all parameters in the same figure).



(c) (5%) Have you noticed any trends in the experiments from (a)–(b)? How does the baseline affect the average reward in Phase 1 and Phase 2, respectively?

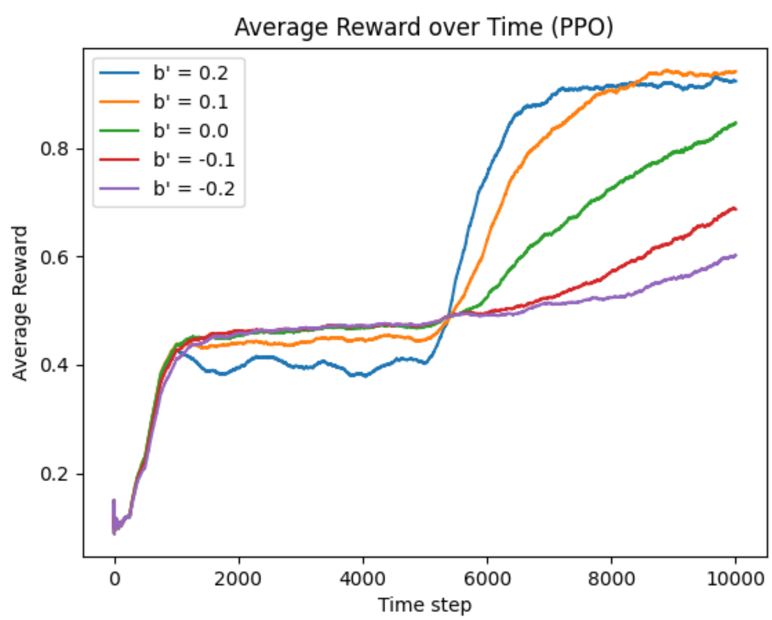
Answer The reasoning is quite similar to the similar question in HW1, just that now higher baseline leads to more exploration. But unlike in EG/BE/IGW where pure exploitation leads to satisfying performance, here, when we set b to be very low (-0.5), the performance is quite bad in both phases. This is probably because setting $b = -0.5$ has a very bad effect: after choosing an action, no matter what reward we get, the algorithm will tend to increase the probability of choosing that action in the next round. Setting $b = 0$ or $b = -0.5$ will make the algorithm totally unable to discover the change point at $t = 5000$.

Adaptive Baseline

(d) (5%) Implement Algorithm 1 with adaptive baseline and, for different values of the extra baseline b' in (6), record in the table below the average reward in Phase 1, Phase 2, and over the entire horizon.

b'	Phase 1	Phase 2	Overall
0.2	0.3837	0.8557	0.6197
0.1	0.4212	0.8241	0.6226
0	0.4370	0.6867	0.5619
-0.1	0.4380	0.5708	0.5044
-0.2	0.4344	0.5308	0.4826

(e) (5%) Plot the running average reward over time for your experiments in (a) (all parameters in the same figure).



(f) (5%) Is there any difference between adaptive baseline and fixed baseline? What are the potential advantages or disadvantages of using adaptive baseline?

Answer The best-performed curve under adaptive baseline seems to be close to the combination of the best-performed ones in Phase 1 and Phase 2 under fixed baselines. The adaptive baseline is able to track the average performance of the algorithm itself and tune baseline on it. Disadvantages include: extra compute and memory, and potentially backfiring instability depending on e.g. the particular problem.

(g) (5%) What is the effect of the extra baseline parameter b'?

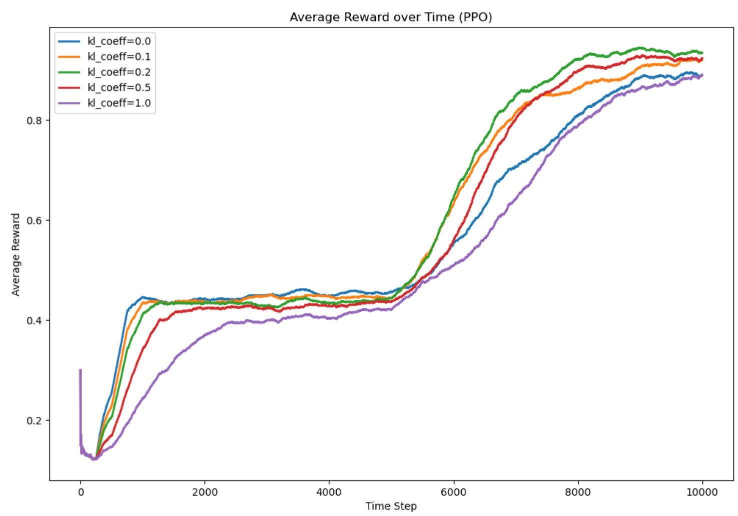
Answer Intuitively, the learner tends to increase the probability of a particular action if its reward is higher than the baseline, and decrease if it is lower than the baseline. With a positive extra baseline b', the learner will more aggressively seek actions that can improve over its current performance. Therefore, a positive extra baseline allows the algorithm to more quickly converge to the new optimal action after the change point.

3.1.2 The Effect of KL Regularization

(h) (5%) Use your best performed baseline setting discovered in Section 3.1.1 with different values of $1/\eta$ in (5). Record in the table below the average reward in Phase 1, Phase 2, and over the entire horizon.

$1/\eta$	Phase 1	Phase 2	Overall
0	0.4289	0.7439	0.5864
0.1	0.4203	0.8033	0.6118
0.2	0.4105	0.8310	0.6207
0.5	0.3919	0.7972	0.5946
1	0.3522	0.7169	0.5345

(i) (5%) Plot the running average reward over time for your experiments in (h) (all parameters in the same figure).



(j) (5%) What is the effect of the KL regularization?

Answer In Phase 1, more KL regularization leads to lower performance and slower convergence, indicating that stronger regularization results in more exploration and slower policy updates. In Phase 2, an intermediate level of KL regularization helps faster discovery of the change, as it provides sufficient exploration without making the updates too slow.

3.2 PG

Algorithm 2 Policy Gradient (also known as REINFORCE)

Default hyperparameters: $N = 8$.
Randomly initialize a policy network π_θ that takes contexts as input and outputs an action distribution.
Let θ_1 be the initial weights for the policy network.
Initialize a baseline network b_ϕ.
for $t = 1, \ldots, T$ **do**
  for $n = 1, \ldots, N$ **do**
    Receive context $x_{t,n}$.
    Sample action $a_{t,n} \sim \pi_{\theta_t}(\cdot|x_{t,n})$
    Receive reward $r_{t,n}$.

$$\theta_{t+1} \leftarrow \theta_t + \lambda \left\{ \frac{1}{N} \sum_{n=1}^{N} \left[\nabla_\theta \log \pi_\theta(a_{t,n}|x_{t,n}) \right]_{\theta=\theta_t} (r_{t,n} - b_{t,n}) \right\}, \tag{8}$$

  where the adaptive baseline is defined as

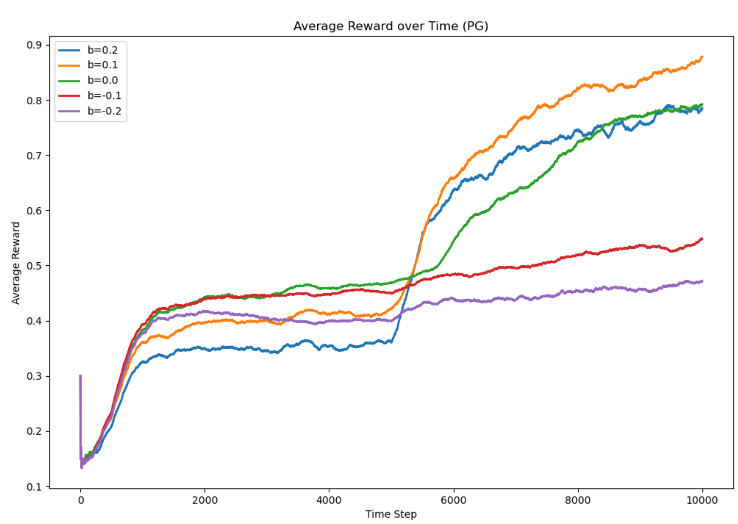
$$b_{t,n} = b_\phi(x_{t,n}) + b'. \tag{9}$$

  Update

$$\phi \leftarrow \phi - \lambda' \nabla_\phi \left[\frac{1}{N} \sum_{n=1}^{N} (b_\phi(x_{t,n}) - r_{t,n})^2 \right]. \tag{10}$$

Because of the log in Eq. (10), the policy network does not need to have the "softmax" layer at the end. That means the network can just output $\log \pi_\theta(a|x)$. It is also totally fine if you add a softmax layer — the two solutions are simply equivalent to each other.

(k) (5%) Implement Algorithm 2 with adaptive baseline and for different values of the extra baseline b' in (9). Record in the table below the average reward in Phase 1, Phase 2, and over the entire horizon.



b'	Phase 1	Phase 2	Overall
0.2	0.3343	0.7094	0.5218
0.1	0.3809	0.7666	0.5737
0	0.4219	0.6755	0.5487
-0.1	0.4183	0.5101	0.4642
-0.2	0.3841	0.4494	0.4168

(l) (5%) Compare the results with those in (d) for PPO. Are the performances of Algorithm 1 and Algorithm 2 different? If so, what do you think is the main cause?

Answer PPO generally achieves better performance. The two algorithms have different optimal tunings: PPO allows for larger N and M, while PG only works with smaller N and with $M = 1$. This is because PPO supports slight off-policy training through importance weighting, while PG requires strictly on-policy training since it couples the gradient of the policy and importance weighting ($\frac{\nabla \pi_\theta(a|x)}{\pi_\theta(a|x)} = \nabla \log \pi_\theta(a|x)$). This flexibility improves the data efficiency of PPO compared to PG.

Another potential reason is the KL regularization in PPO, which effectively regularizes updates in the probability space rather than the parameter space as in PG. The benefit of this — avoiding slow recovery of low-probability actions — has been discussed in class (see Pages 54–55 in the slides).

4 Survey

(5%) Leave any feedback for the course.

A Lagrange Multiplier

Theorem 1. *Let $\Omega \subset \mathbb{R}^d$ be a non-empty region and let $f, g : \Omega \to \mathbb{R}$ be two functions defined on it. Furthermore, let*

$$\mathcal{F} = \{x \in \Omega : \ g(x) = 0\} .$$

Let $x^\star = \mathrm{argmax}_{x \in \mathcal{F}} f(x)$, i.e., $x^\star$ is a maximizer of f in Ω under the constraint $g(x) = 0$. If the following are true:

- *$x^\star$ is not on the boundary of Ω,*
- *f, g are continuously differentiable in the neighborhood of $x^\star$,*
- *$\nabla g(x^\star) \neq 0$,*

then there exists $\lambda \in \mathbb{R}$ such that

$$\nabla f(x^\star) + \lambda \nabla g(x^\star) = 0. \tag{11}$$