Week 8 Precept

Catherine Ji

March 23, 2025

1 Introduction

Today, we'll review PPO and tie up some loose ends. PPO consists of (ask):

- Advantage estimate of the policy gradient.
- Importance sampling to make Actor-Critic off-policy. This is so we have access to a replay buffer, and don't have to ONLY train on-policy, which is super inefficient.
- To combine bias-variance tradeoff in Advantage estimation: Generalized Advantage Estimation (GAE). Aka, average between varying degrees of monte carlo and TD estimates of advantage.
- Trust region: bounding the difference between π_{old} and π by KL-divergence! Note that importance sampling is NOT an unbiased estimator of the policy gradient for target θ – at the same time, we want π_{old} and π to be different (off-policy), so we can't just use the unbiased estimator.

2 What is a discounted state-occupancy measure?

First, though, I'd like to make sure we're all on the same page about discounted state-occupancy measures. We can always rewrite PG estimates over trajectories in terms of the discounted state-occupancy measure:

We can replace any $\pi_\theta(\tau)$ with the expectation over the trajectory-induced probability distribution over states, with γ weightings to "simulate" absorbing states (probability of "death" is $1-\gamma$ per timestep). Below, we relabel $\sum_{t'=t}^{T} \gamma^{t'-t} r(s_{t'}, a_{t'})$ as $G_t = G(s_t, a_t)$).

Effectively, we're saying to forget about iterating along the trajectories – if we follow the policy for a ton of samples, we will effectively sample from a state distribution *induced* by the policy called the **dicounted state occupancy measure**, and can abstract away annoyances of computing policy-gradient transition-by-transition:

$$\theta_{i+1} \leftarrow \theta_i + \xi \mathbb{E}_{s_t \sim P_\gamma^\pi(\cdot), a_t \sim \pi(\cdot|s_t)}[\nabla_\theta \log \pi_\theta(a \mid s) G(s,a)].$$

Explicitly, the **discounted state occupancy measure** is as follows (prefactor is just for normalization)

$$P_\gamma^\pi(s) \triangleq \frac{1-\gamma}{1-\gamma^{T+1}} \sum_{t=0}^{T} \gamma^t p^\pi(s_t = s),$$

where we define (written out explicitly for clarity):

$$
\begin{aligned}
p^\pi(s_t = s) &= \text{(Probability that we're in state s at time t by following policy from start distribution)} \\
&= \sum_{(s_0, a_0, \cdots s_{t-1}, a_{t-1}) \in (\mathcal{S} \times \mathcal{A})^t} p(s_t = s \mid s_{t-1}, a_{t-1}, s_{t-2}, a_{t-2}, \cdots) p^\pi(s_{t-1}, a_{t-1}, s_{t-2}, a_{t-2}, \cdots) \\
&= \sum_{(s_0, a_0, \cdots s_{t-1}, a_{t-1})} p(s_t = s \mid s_{t-1}, a_{t-1}) p^\pi(s_{t-1}, a_{t-1}, s_{t-2}, a_{t-2}, \cdots) \\
&= \sum_{(s_0, a_0, \cdots s_{t-1}, a_{t-1})} p(s_t = s \mid s_{t-1}, a_{t-1}) p^\pi(s_{t-1}, a_{t-1} \mid s_{t-2}, a_{t-2}) p^\pi(s_{t-2}, a_{t-2}, \cdots) \\
&= \sum_{(s_0, a_0, \cdots s_{t-1}, a_{t-1})} p(s_t = s \mid s_{t-1}, a_{t-1}) p(s_0) \prod_{i=0}^{t-2} p^\pi(s_{i+1}, a_{i+1} \mid s_i, a_i) \\
&= \sum_{(s_0, a_0, \cdots s_{t-1}, a_{t-1})} p(s_t = s \mid s_{t-1}, a_{t-1}) p(s_0) \prod_{i=0}^{t-2} p(s_{i+1} \mid s_i, a_i) \pi(a_{i+1} \mid s_{i+1})
\end{aligned}
$$

A great question fom lecture today ("I thought dynamics are Markovian, so why do we use this object?"): yes, we can indeed write/break down P_γ^π in terms of the transitions, policies, etc. However, this is all quite taxing to write out as you can see – it's also easier to think about (and do theory over) the discounted state occupancy measure as a way to sample states from the policy-induced trajectory distribution *without* necessarily needing to (mentally) iterate timestep-by-timestep through individual, sampled trajectories.

3 TRPO and PPO objective

Explcitly, our loss objective (WHEN WE UPDATE THE ACTOR, note we have a separate critic updated via Expected SARSA or similar off-policy method) is

$$
\mathcal{L}(\theta) = \mathbb{E}_{s \sim P_\gamma^{\pi_{\text{old}}}(\cdot), a \sim \pi_{\text{old}}(\cdot \mid s)} \left[\frac{\pi_\theta(a \mid s)}{\pi_{\theta_{\text{old}}}(a \mid s)} A^{\pi_{\theta_{\text{old}}}}(s, a) \nabla_\theta \log \pi_\theta(a \mid s) \right].
$$

Here, $A^{\pi_{\text{old}}}$ tells us how good the action a is in state s according to the old policy, and directs our gradient for changing π_θ. This reasoning may sound familiar – we can actually think of the old policy as specifying our "target" in some sense!

You may ask – why use the old policy for the Advantage function? In fact, using Q^{π_θ} is is done in another algorithm (Off-Policy Policy Gradient (Degris, White, Sutton 2012) with also own set of theoretical guarantees, but generally, TRPO and PPO has better empirical performance because we no longer have a moving "target".

Great! So if the objectives are the same... then what are the real differences between TRPO and PPO? The key is in the **constraints** on policy updates.

- **TRPO:** TRPO constrains the KL-divergence between the old and new policy. So you are solving a constrained optimization problem, which is hard and generally annoying because the KL-divergence is also nontrivial to compute (non-convex, among other things).
- **PPO:** PPO, on the other hand, clips the policy update itself in a way that, very loosley and not theoretically computed to my knowledge (but I am sure someone out there can write down that PPO clip is some kind of approximation to the TRPO KL-divergence constraint), tries to keep the DISTRIBUTIONS of the old and new policy close.

That's it! That's the difference! So don't get too bogged down in the details – the key is that PPO is a more practical,

easier-to-implement version of TRPO where the simplification is made in the policy distance constraint.

4 Trust regions: where do they come from?

Another loose end I wanted to tie up: let's take a step back. Why trust regions? Where does the KL Divergence object come from? Below, we summarize at a very high level the proof Schulman et al. wrote for the TRPO paper (Schulman et al. 2017), which produces the KL-divergence term.

4.1 Preliminaries and Surrogate Objective

Some preliminaries, all from the same TRPO paper:

- **"Value function" $\eta(\pi)$:** We define a "value function" $\eta(\pi)$ over the **policy-induced discounted state occupancy measure**. The only difference between this and the value function is that the value function is defined over starting states – meanwhile, $\eta(\pi)$ assumes we are drawing our starting state s_0 from some MDP-specific start distribution $p(s_0)$ (and roll out policy accordingly from this state). **This object is equivalent to the policy gradient objective $R(\tau)$ that we considered before!!!** But we're using this instead to express our objects with the discounted state occupancy measure and the policy, only.

- **Rewrite $\eta(\tilde{\pi})$ in terms of the Advantage function (this is cool!):** For any arbitrary $\tilde{\pi}$ and π, we can always write

$$\eta(\tilde{\pi}) = \eta(\pi) + \mathbb{E}_{s\sim P_\gamma^{\tilde{\pi}}(\cdot), a\sim\tilde{\pi}(\cdot|s)}\left[A^\pi(s,a)\right].$$

  There's a nice proof in Appendix A of their paper. But intuitively, the idea is this: advantage functions are neat because they tell you how much better action a at a given state is relative to your current policy – the effect of a is "isolated" in some sense. So, if you're sampling actions a from a *different* policy-induced state distribution (call this distributon $\tilde{\pi}$)and plugging these sampled actions and states into A^π (note that this is over π!), you are, in essence, teasing out the "value function" of $\tilde{\pi}$ MODULO the original value function over π! After all, the sauce of the policy is in the local actions – and Advantage functions isolate/insulate the effect of actions.

- **$L_\pi(\tilde{\pi})$, local approx. to $\eta(\tilde{\pi})$ USING P_γ^π:** This is just a quick extension of the above. We can approximate $\eta(\tilde{\pi})$ locally around π by kind of "Taylor expanding" around the discounted state occupancy measure induced by π – remember that we explicitly are trying to compute our advantage function using a "target" that is old, aka our old policy, and thus only have access to the distribution of states induced by the old policy. Concretely,

$$L_\pi(\tilde{\pi}) = \eta(\pi) + \mathbb{E}_{s\sim P_\gamma^{\pi}(\cdot), a\sim\tilde{\pi}(\cdot|s)}\left[A^\pi(s,a)\right] \approx \eta(\tilde{\pi}).$$

  The only difference is a really tiny tilde over the π in the discounted state occupancy measure – but this is actually a really important difference!

Note that this looks a ton like our **surrogate objective** – that's because it is! But because we're sampling from the old and new policies, instead we can just do the important weighting and only sample from the old policies (because we have access to the π and $\tilde{\pi}$ distributions), so

$$L_\pi(\tilde{\pi}) = \eta(\pi) + \mathbb{E}_{s\sim P_\gamma^{\pi}(\cdot), a\sim\pi(\cdot|s)}\left[\frac{\tilde{\pi}(a\mid s)}{\pi(a\mid s)}A^\pi(s,a)\right] \approx \eta(\tilde{\pi}).$$

To double check that this is a true surrogate objective, we note that (using π_θ and $\pi_{\theta_{\text{old}}}$ for clarity):

$$\text{grad}_\theta \, L_{\pi_{\text{old}}}(\pi_\theta)|_{\theta=\theta_{\text{old}}} = \text{grad}_\theta \, \eta(\pi_\theta)|_{\theta=\theta_{\text{old}}}.$$

The proof is left as a worthwhile exercise (hint: policy gradient derivation, which can also be done using the discounted state occupancy measure formulation... the result with the discounted state occupancy measure is super satisfying!). Thus, this means that the surrogate objective is a first-order (in $|\theta - \theta_{\text{old}}|$) approximation to the true objective... but we still don't know how big of a step to take in parameter space!

For that, we'll need a bound on just how far off the surrogate objective is. Now, we have all the pieces to make Schulman et al.'s main statement that answers this question, which reveals the trust region constraint.

Theorem 1 (Bound on updated $\eta(\tilde{\pi})$ from TRPO paper (Schulman et al. 2017))
Let π_{new} be our new policy, and π_{old} be some old policy. Then, in the tabular setting (probably can be extended to continuous as well, but the key is assuming no function approximation errors in our advantage), we have that

$$\eta(\pi_{\text{new}}) \geq L_{\pi_{\text{old}}}(\pi_{\text{new}}) - C D_{KL}^{\max}(\pi_{\text{old}} \mid \pi_{\text{new}}),$$

where C is some constant defined in the paper, and the max is taken with respect to the states. The proof is in the Appendix of the paper. The first proof is quite clean, the second is a bit less readable for those unfamiliar with perturbation theory.

Voila! We have a clear reason to bound our trust region now – in the theoretical setting, we can guarantee that our surrogate will be off by some KL-divergence penalty. So, thus, we constrain the KL-divergence between old and new!

5 PPO in its full form: bringing it all together

Let's put all the moving parts together! Incoming algorithmic block.

Algorithm 1 Proximal Policy Optimization (PPO)

Require: Policy parameters θ, value parameters ϕ, clipping parameter ϵ, GAE parameter λ, discount factor γ, learning rates α_π, α_v

for iteration = 1,2,... **do**

  Collect set of trajectories $\mathcal{D} = \{(s_t, a_t, r_t, s_{t+1})\}$ by running policy π_θ

  Compute advantages $\hat{A}_t$ using GAE:

$$\delta_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t), \quad \hat{A}_t = \sum_{l=0}^{T-t} (\gamma\lambda)^l \delta_{t+l}$$

  Compute discounted returns $\hat{R}_t = \hat{A}_t + V_\phi(s_t)$

  for epoch = 1,2,...,K **do**

    Shuffle and split $\mathcal{D}$ into minibatches

    for each minibatch **do**

      Compute probability ratio:

$$w_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}$$

      Policy loss (note that this loss's gradient is the same (modulo clipping and sign) as the surrogate objective's gradient!):

$$L^{\text{clip}}(\theta) = -\mathbb{E}\Big[\min\big(w_t(\theta)\hat{A}_t,\ \text{clip}(w_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\big)\Big]$$

      Value loss (note this is totally separate from Actor update):

$$L^V(\phi) = \mathbb{E}\big[(V_\phi(s_t) - \hat{R}_t)^2\big]$$

      Total loss:

$$L(\theta, \phi) = L^{\text{clip}}(\theta) + c_1 L^V(\phi) - c_2\mathbb{E}[\mathcal{H}(\pi_\theta(\cdot|s_t))]$$

      Update parameters:

$$\theta \leftarrow \theta - \alpha_\pi \nabla_\theta L, \quad \phi \leftarrow \phi - \alpha_v \nabla_\phi L^V$$

    end for

  end for

  $\theta_{\text{old}} \leftarrow \theta$

end for