September 28

1 Intro: Proximal Policy Optimization

Proximal Point Method Going beyond gradient descent, proximal point methods offer a more general framework for optimization. Starting from an initialization θ^0, they perform iterative updates of the form:

$$\theta^{k+1} = \arg\max_\theta \left\{ R(\theta) - \beta D(\theta, \theta^k) \right\},$$

where, at iteration k, the next point θ^{k+1} is obtained by maximizing the objective while penalizing deviation from θ^k. Here, $D(\theta, \theta')$ is a discrepancy or distance measure between θ and θ'. The default choice is $D(\theta, \theta') = \frac{1}{2}\left\|\theta - \theta'\right\|^2$, though it is common to use KL divergence in policy optimization as we show below.

Each update from θ^k to θ^{k+1} requires solving an optimization problem, making the proximal point method a double-loop algorithm. Since this inner problem is often intractable, it is typically approximated using practical methods such as gradient descent or proximal gradient descent.

Different approximations lead to different algorithmic variants.

Proximal Gradient Descent Applying a first-order Taylor approximation to the objective function $R(\theta)$ gives:

$$\theta^{k+1} = \arg\max_\theta \left\{ \nabla R(\theta^k)^\top (\theta - \theta^k) - \beta D(\theta, \theta^k) \right\}.$$

This yields the *proximal gradient descent* method.

Preconditioned Gradient Descent Consider a quadratic proximal penalty:

$$D(\theta, \theta^k) = \frac{1}{2}(\theta - \theta^k)^\top J(\theta^k)(\theta - \theta^k),$$

where $J(\theta^k)$ is a positive definite preconditioning matrix chosen by the user. In this case, proximal gradient descent admits a closed-form solution:

$$\theta^{k+1} = \theta^k + J(\theta^k)^{-1}\nabla R(\theta^k).$$

This is known as *preconditioned gradient descent.*

Remark Assume we use gradient descent to solve the inner loops of proximal methods. The difference from vanilla gradient descent then lies in how the *reference point* is updated. To see this, let us consider the following generic update rule:

$$\theta^{k+1} = \arg\max_\theta \Big\{ \nabla R(\theta^k)^\top (\theta - \theta^k) - \beta D(\theta, \theta^{\mathrm{ref},k}) \Big\},$$

where $\theta^{\mathrm{ref},k}$ is the reference point used to anchor the update at iteration k. The choice of $\theta^{\mathrm{ref},k}$ governs the behavior of the algorithm:

- If $\theta^{\mathrm{ref},k} = \theta^k$, the update reduces to *proximal gradient descent*, where the reference point is updated as fast as the iterate itself.
- If the reference point is updated only once every m iterations, i.e. $\theta^{\mathrm{ref},k} = \theta^{m\lfloor k/m \rfloor}$, then each outer iteration corresponds to running m steps of a proximal gradient algorithm to approximately solve the proximal point update.
- If $\theta^{\mathrm{ref},k}$ evolves more slowly, for example as an exponential moving average of past iterates, the algorithm takes more conservative steps, effectively stabilizing the optimization.

In short, the proximal point framework interpolates between fast-updating methods like gradient descent and slower, more implicit schemes, depending on how quickly the reference point is advanced.

Proximal Policy Optimization (PPO) Applying the proximal point method and using importance sampling for reward estimation, we have

$$\theta^{k+1} = \arg\max_{\theta} \left\{ \mathbb{E}_{\mathcal{D}} \left[\frac{\pi^{\theta}(a|s)}{\pi^{\mathtt{data}}(a|s)} r(a,s) \right] - \beta D(\pi^{\theta} \mid\mid \pi^{\theta^k}) \right\}.$$

We discuss two key design choices:

- *Penalty Function.* Typically, the discrepancy D is taken to be the KL divergence:

$$D(\pi^{\theta} \mid\mid \pi^{\theta^k}) = \mathbb{E}_{s\sim p_0}[\mathrm{KL}(\pi^{\theta}(\cdot|s) \mid\mid \pi^{\theta^k}(\cdot|s))].$$

However, various penalty functions have been used. We review these in the sequel.

- *Clipping Function.* Perhaps the most important trick in PPO is the clip function:

$$\theta^{k+1} = \arg\max_{\theta} \left\{ \mathbb{E}_{\mathcal{D}} \left[\mathcal{C}(w_{\theta}(a,s), r(a,s)) \right] - \beta D(\pi^{\theta} \mid\mid \pi^{\theta^k}) \right\},$$

where $w_{\theta}(a,s) = \frac{\pi^{\theta}(a|s)}{\pi^{\mathtt{data}}(a|s)}$, and $\mathcal{C}$ is defined as

$$\mathcal{C}(w,r) = \min(wr,\ \mathtt{clip}(w, [1-\epsilon, 1+\epsilon])\, r).$$

In the following, we discuss the implications of the design choices for the penalty and clip functions, respectively.