1 Lecture 4: Model Free Control and Function Approximation

2 Lecture 5: Policy Gradient and Search

Policy gradient/ search is influential in NLP/ Proximal Policy Optimization (training GPT). The core idea:

Intuition of Gradient Research

Approximate $V^\pi(s) \approx V_W(s)$ and $Q_w(s,a) \approx Q^\pi(s,a)$ by adjusting weight w.
Policy gradient: rather than generating policy from value (ϵ-greedy), directly parametrize policy with θ, i.e.

$$\pi_\theta(s,a) = \mathbb{P}[a|s;\theta]\text{: optimize } V(\theta) \text{ to find policy } \pi$$

The brief classification of policy gradient is as follows:

		Value-based	Policy-based
Actor-critic			
Value function	learned	not present	learned
Policy	implicit (ϵ-greedy) learned	learned	

Instead of deterministic/ ϵ-greedy policies, need to focus heavily on **stochastic** for direct policy search!

- Repeated Trials, e.g. In rock paper scissors (of many rounds), deterministic policy is easily exploited by adversary.
- Boundary Condition, e.g. In gridworld, bound to only move one direction (else get stuck/ traverse for long time for slow convergence).

In short, poliy objective functions have the following intuition:

Policy Objective Summary

- Goal: Given policy $\pi_\theta(s,a)$, find best parameter θ.
  Inherently, an optimization of $V(s_0, \theta)$.
- Purpose: Measure quality for policy π_θ with policy value at start state s_0.
- Works for: both episodic/ continuing and infinite horizons.

2.1 Gradient Free Policy Optimization

They are great simple baselines.

- Examples: Hill Climbing, Genetic Algo (evolution strategies, cross-entropy method, covariance matrix adaption)
- Known for decades but embarrassingly well: rivals standard RL techniques!
- Advantages: Flexible for any policy parameterization, easily to parallelize
  Disadvantage: Less sample efficient (ignores temporal structure)

2.2 Policy Gradient

This section focuses on gradient descent; other popular algos include conjugate gradient and quasi-newton methods.
Usually assume **Episodic MDPs** for easy extension of objectives. The method:

Define $V(\theta) = V(s_0, \theta)$, i.e. the value fucntion depending on policy parameters. Then:

- Search the local maximum in $V(s_0, \theta)$ with gradient increments:

$$\Delta\theta = \alpha \nabla_\theta V(s_0, \theta) = \alpha \begin{pmatrix} \dfrac{\partial V(s_0, \theta)}{\partial \theta_1} \\ \vdots \\ \dfrac{\partial V(s_0, \theta)}{\partial \theta_n} \end{pmatrix}$$

- Assumption: π_θ differentiable (and known gradient $\nabla_\theta \pi_\theta(s, a)$)
- We can rewrite policy value $V(s_0, \theta)$ in the following ways:
  1. **Visited States and Actions**: $\mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T} R(s_t, a_t); \pi_\theta, s_0\right]$
  2. **Weighted Average of Q-values by Actions**: $\sum_a \pi_\theta(a|s_0) Q(s_0, a, \theta)$
  3. **Trajectories Sampled using π_θ**: $\sum_\tau P(\tau|\theta) R(\tau)$

In particular, it is of interest to consider writing $V(s_0, \theta)$ in trajectory form:

To find the best policy parameter θ, we consider

$$\arg\max_\theta V(\theta) = \arg\max_\theta \sum_\tau P(\tau; \theta) R(\tau)$$

Taking gradient,

$$\begin{aligned} \nabla_\theta V(\theta) = & \nabla_\theta \sum_\tau P(\tau; \theta) R(\tau) \\ & \sum_\tau \nabla_\theta P(\tau; \theta) R(\tau) (R \text{ being indep of } \theta) \\ & \sum_\tau \frac{P(\tau; \theta)}{P(\tau; \theta)} \nabla_\theta P(\tau; \theta) R(\tau) \\ & \sum_\tau R(\tau) P(\tau; \theta) \nabla_\theta \log P(\tau; \theta) \qquad \text{(log-likelihood)} \end{aligned}$$

Approximate in practice using m sample trajectories under π_θ:

$$\nabla_\theta V(\theta) \approx \hat{g} = \frac{1}{m} \sum_{i=1}^{m} R(\tau^{(i)}) \nabla_\theta \log P(\tau^{(i)}, \theta)$$

But trajectories can be decomposed into states and actions:

$$\begin{aligned} \nabla_\theta \log P(\tau^{(i)}; \theta) = & \nabla_\theta \log \left[\mu(s_0) \prod_{t=0}^{T-1} \pi_\theta(a_t|s_t) P(s_{t+1}|a_{t+1}, s_{0:t}, a_{0:t}) \right] \\ & \sum_\tau \nabla_\theta \log \pi_\theta(a_t|s_t) \end{aligned}$$

Here

- We call $\sum_\tau \nabla_\theta \log \pi_\theta(a_t|s_t)$ the **score function**.
- the initial state $\mu(s_0)$ is constant; dynamics model $P(s_{t+1}|a_{t+1}, s_{0:t}, a_{0:t})$ is invariant to θ.
- In other words, no dynamics model is required to approximate the policy parameter θ.

Questions

1. Why trajectory form is practical ("better in training")?
2. Why is log-likelihood ratio important here? What does it enable?

2.3 VFA under TDL

3 Lecture 5-6: Remedies to Reduce Variance in Policy Gradient