RL W-7 (17/3/2025).

Before Jumping Into Policy Grad...

GPI
- poleval → given policy $\pi$, what is $V(\pi)$?
- polimprov → modify $\pi^k$ such that $V(\pi^{k+1}) \geq V(\pi^k)$ "better"

↕

Generalized Inexact

$e_{PB}$

$e_{ms}$: semi-grad → in RL → policy gradient.

e.g. via grad ascent method.

grad: of what? value. w.r.t its param

ascent: why? to maximize i.e.,

$\nabla V_x(\pi) = \frac{\partial V_x(\pi)}{\partial \pi(\cdot)}$ ; policy!

$\pi_x^* \in \text{argmax}_{\pi \in \Pi_s} V_x(\pi)$

strictly "$\in$"

polgrad in direct policy parameterization

→ hard to use → not beneficial

FAKULTAS ILMU KOMPUTER

As a remedy for direct policy parameterization:

→ Use explicit policy parameterization!

↳ where we introduce param vector $\theta \in \Theta : \mathbb{R}^{\dim(\theta)}$.

How?

So: $V(\pi(\theta)) \stackrel{\text{def.}}{=} V(\theta)$.

Example:

Let $\pi \in \Pi_{SR}$

Set $\pi \sim \text{Gaussian}(\mu=0, \sigma=\theta)$.

proven empirical success

↑

Why? Now it's becoming best practice & SOTA!

① utilize high-dimension representation for policy
e.g., nets → learning $\sigma$.

② Encode levels of exploitation
↳ share the similar idea with $\varepsilon$-greedy.

③ ?

# The Gradient Ascent Rule / For inexact policy improvement

$$\theta^{k+1} \leftarrow \theta^{k} + \alpha \cdot \nabla V_x(\theta)\Big|_{\theta=\theta^k}$$

- $\alpha$: positive step size
- $\nabla V_x(\theta)$: evaluated using $\theta^k$
- $k$: iteration index

Assume:

$V_x(\theta)$ is differentiable.

Example:

$f(x) = 2x^2 \rightarrow f'(x) = 4x\Big|_{x=1}$

$4(1) = 4$,

Deterministic → Not differentiable

require smoothness → Stochastic!

How?

$\theta$ : init, e.g., randomly.

$\alpha$ : init e.g., $10^{-3}$

$\nabla V_x(\theta) = ?$ Big Question

Answer:

Sutton's Polgrad Theorem!

set $x \leftarrow$ gain.

$V_g = \rho^{\pi} \cdot r_{\pi}$ → matrix form

$\rho^{\pi}$ ... $r_{\pi}$

$$\nabla V_g = \nabla \left\{ \sum_{s \in S} \sum_{a \in A} r(s,a)\, \rho^{\pi}(s)\, \pi(a|s;\theta) \right\}$$ (by def of gain).

$$= \sum_{s} \sum_{a} r(s,a) \nabla \left\{ \rho^{\pi}_{\pi(\theta)}(s)\, \pi(a|s;\theta) \right\}$$

$\pi(\theta)$: $\rho^{\pi}$ depends on $\theta$, as $\pi$ depends on $\theta$

Note that $\rho^{\pi}$ depends on $\theta$, as $\pi$ depends on $\theta$

FAKULTAS ILMU KOMPUTER

Continuing on $\nabla V_\theta$ ....

Apply derivative's product rule

$$= \sum_s \sum_a r(s,a) \left\{ \nabla p^*_\pi \cdot \pi + p^*_\pi \cdot \nabla \pi \right\}.$$

We don't know $p^*$ → Not sampling friendly!

Thus, we need score function!

$$\nabla \log \pi(a|s;\theta) = \frac{\nabla \pi(a|s;\theta)}{\pi(a|s;\theta)} \quad \left.\right\} \text{diff of log.}$$

$$= \sum_s \sum_a r(s,a) \left\{ p^*_\pi \nabla \log p^*_\pi \, \pi + p^*_\pi \cdot \pi \cdot \nabla \log \pi \right\}$$

(by score function)

$$= \sum_s \sum_a p^*(s) \pi(a|s) \cdot r(s,a) \left\{ \nabla \log p^* + \nabla \log \pi \right\}$$

Sampling friendly, but in RL $p^*$ is unknown to the agent!

Fun fact: Sutton spent days to solve this.

FAKULTAS ILMU KOMPUTER

Cont...

→ There's $p^\star$ still, but it's becoming sample distribution $s \sim p^\star(\cdot)$.

$$= \sum_s \sum_a p^\star(s) \cdot \pi(a|s) \cdot q_b^\pi(s,a) \cdot \nabla \log \pi(a|s;\theta)$$

$q_b^\pi(s,a)$ → encode $\nabla \log p^\star$ and $r(s,a)$.

$$= \mathbb{E}\left[ q_b^\pi(S,A) \cdot \nabla \log \pi(A|S;\theta) \right]$$ → proof?

$S \sim p^\star(\cdot)$ → unknown but appears under $\mathbb{E}$

$A \sim \pi(\cdot|s)$

RV.

a.k.a. mean $\mu \approx$ (approximated by) sample mean, $\bar{\mu}$,

which is unbiased, i.e., $\mathbb{E}[\bar{\mu}] = \mu$.

hence stochastic gradient ascent.

stochastic: sample mean

gradient ascent: unbiased.

run unbiased approx of $\nabla V_x(\pi)$

hence SGA

hence in practice, we simply sample from $p^\star_{\pi(\theta)}$ → exist at $t \gtrsim t_{mix}$

As a programmer, likely $t_{mix}$ can be computed in small envs, but not in large scale, thus it's becoming hyperparams!

At $t < t_{mix}$, $s \sim p^{t} \neq p^{\pi}$,

hence it's **biased** approx of $\nabla V_x(\pi)$.

↳ semi-SGA

- Code:

gymnasium → env library

stable_baseline → algo library.

We can also create custom envs!

Cont...

Issue: sample mean of $\mathbb{E}[q \nabla \log \pi]$ is unbiased but "generally" has high variance as:

involves $p^*$ and $\Omega \rightarrow$ 2 RVs!

as a remedy.... ~~fast fast~~, e.g.,

(1) Baseline control variate (BCV)
(2) Actor-critic

} can complement each other!

→ (1) BCV: goal: reduce variance ~~different~~ from bias-error

different from b in $V_b$.

How: use a baseline independent of action, e.g., $V_x(s)$

Hence,

$$\nabla V_g(\theta) = \mathbb{E}\left[\left\{ q_b^\pi(S,A) - V_b(s) \right\} \nabla \log \pi(A|S;\theta)\right]$$

advantage (state-action) function $a_b^\pi(s,a)$.

baseline

~~Fun fact~~ $= \mathbb{E}[\delta_v(S,A,S')]$

Fun fact:

∃ theorem:

$$\partial(s,a) = \mathbb{E}\left[\underbrace{r(s,a) - V_g + V_b(s') - V_b(s)}_{TD: \delta_V} \,\middle|\, s,a\right]$$

$s' \sim p(\cdot|s,a)$

Says TD is unbiased estimate for $\partial$!

Thus:

$$= \mathbb{E}\left[\delta_V(s,A,s')\nabla \log \pi(A|s;\theta)\right]$$

Using sample of $(s,a,s')$:

$$\approx \delta_V(s,a,s')\nabla \log \pi(a|s;\theta)$$

② Actor Critic: reduce variance using policy value approximator

↓

introduce bias!

$w$ : parametric $w \in W$

aka critic! forms $\hat{V}_b(s; w)$

actor refers to: ~~$\pi(a|s;\theta)$~~, $\pi(\theta)$

which used in

④ $\delta_v$ for approx $\nabla$

for approx $\nabla V_g(\theta)$

Here we apply both:

① baseline.

② Actor critic

# RL Week-7-2

Policy Param for Diseret Action.

(focus : finite MDP).

↳ S, A finite

e.g., using Gibbs / Boltzmann / Categorical Distribution.

Softmax.

$$\pi(a|s;\theta) = \frac{\exp(\theta^T \phi(s,a))}{\sum_{a' \in A} \exp(\theta^T \phi(s,a'))} \quad ; \forall (s,a) \in S \times A.$$

↓ conditional probability over action.

Recall:

$\pi : S \to \Delta^{|A|}$ or $\pi : S \times A \to [0,1]$.

e.g., $s^0 \to$ | 0.3 | 0.6 | 0.1 |

# Discounted Reward - Policy Gradient -

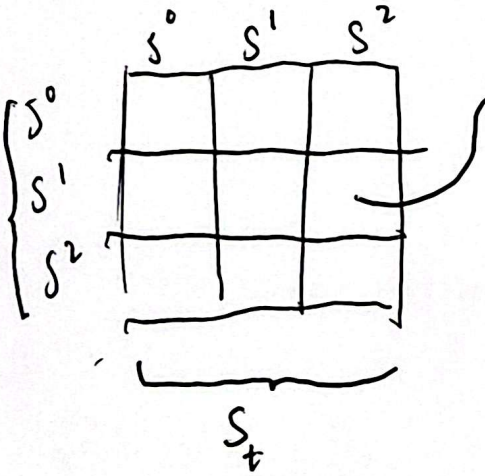
Recall that:

$V_g(\pi) = P_\pi^\star r_\pi$ where $P_\pi^\star = \lim_{t\to\infty}$

$V_\gamma(\pi) = P_\pi^\gamma \cdot r_\pi$

where

$$P_\pi^\gamma = \lim_{t_{max}\to\infty} \sum_{t=0}^{t_{max}-1} (\gamma \cdot P_\pi)^t$$




for each cell:

$= P_\pi^\gamma(s_t \mid s_0)$

in $t$ timesteps

$$= \lim_{t_{max}\to\infty} \sum_{t=0}^{t_{max}-1} \gamma^t P_\pi^t(s \mid s_0)$$

is $P_\pi^\gamma$ a distribution?

↳ if we sum the elements in a row, it'll be 1, e.g.

$$\sum_{s\in S} \left\{P_\pi^\gamma(s \mid s_0)\right\} = 1.$$

$$\sum_{s\in S} \left\{\lim_{t_{max}\to\infty} \sum_{t=0}^{t_{max}-1} \gamma^t \cdot P_\pi^t(s \mid s_0)\right\}$$

$\rho_\pi^\gamma$ is improper distribution, as $\frac{1}{1-\gamma} \neq 1$

Then, how's next?

Recall that:

$V_\gamma$ depends on initial state → introduce complexity c.f. $\nabla V_\gamma(\pi)$

$$\nabla V_\gamma(\pi, s_0) = \sum_s \rho_\pi^\gamma(s|s_0) \sum_{a \in A} q_\gamma^\pi(s,a) \cdot \nabla \pi(a|s;\theta)$$

not sampling friendly as $\rho_\pi^\gamma$ is not a distribution.

Thus, we introduce $\frac{1-\gamma}{1-\gamma}$:

$$\nabla V_\gamma(\pi, s_0) = \frac{1}{1-\gamma} \sum_s (1-\gamma) \rho_\pi^\gamma(s|s_0) \cdot \sum_{a \in A} q_\gamma^\pi(s,a) \nabla \pi(a|s;\theta)$$

proper distribution!

i.e. Geometric Distribution ($p = 1-\gamma$)

Q: How to sample from that distribution?

We can also scale the gradient; just like what $\alpha$ (learning rate) do:

$$(1-\gamma) \cdot \nabla V_\gamma(\pi, s_0) = \sum (1-\gamma) \rho_\pi^\gamma(s|s_0) \sum \ldots$$

# Difference: ① State Distribution:

avg rew

$S \sim P_\pi^\star$

Disc. rew.

$S \sim Geo(p = 1-\gamma)$

↓

number of trials until success, i.e. ~~terminates~~ terminates after $t+1$

## ② Dependency on $S_0$

avg rew

$$\pi^* = \underset{\pi \in \Pi}{\operatorname{argmax}} \, V_g(\pi)$$

uniform optimal w.r.t. $\mathring{p}$.

discounted rew

$$\pi^*_\gamma = \underset{\pi \in \Pi}{\operatorname{argmax}} \, \mathbb{E}_{S_0 \sim \mathring{p}}\left[V_\gamma(\pi, S_0)\right]$$

$\mathring{p}$ → init state dist.

Interpretation:

$\pi^*_\gamma$ depends on $\mathring{p}$ ~~→ how~~

↓

non_uniform optimal

w.r.t $\mathring{p}$, e.g.;

$\mathring{p}_{(1)}$ = | $s^0$ | $s^1$ | $s^2$ |
|---|---|---|
| 1 | 0 | 0 |

$\mathring{p}_{(2)}$ = | 0 | 1 | 0 |
|---|---|---|

How To Sample From $\delta \sim Geo(p = 1-\gamma)$?

See Paper. . . . —

$$\nabla V_\gamma(\pi, s_0) = \frac{1}{1-\gamma} \sum_{s \in S} (1-\gamma) \cdot P_\pi^\gamma \sum_{a \in A} \ldots$$

$$= \frac{1}{1-\gamma} \mathbb{E}[\ \ldots\ ].$$

$$\delta \sim (1-\gamma) \cdot P_\pi^\gamma.$$

$$A \sim \pi.$$

How to sample states from $\underbrace{(1-\gamma) \cdot P_\pi^\gamma}_{Geo(p=1-\gamma)} = \tilde{P}_\pi^\gamma$?

① Sample a length of xprmt-episode from Geo(1-γ)

num of steps = length (xprmt-episode).

$L \sim Geo(1-\gamma)$.

say $l = 20$ steps

② Run an xprmt-episode $t=0 \rightarrow t=20-1$.

③ Thm $s_{t=19} \sim \tilde{P}_\pi^\gamma$ → we throw away all state samples from $t=0$ to $t=18$

↳ Not sample efficient!

In practice, people use all state $\hat{P}_\pi^\gamma$ ! → biased → semi-SGA!