Approximate Policy Iteration and Variants

Chen-Yu Wei

Policy Iteration

For $k = 1,\ 2, \ldots$

Calculate $Q^{\pi_k}(s,a) \quad \forall s,a$

$\pi_{k+1}(s) = \underset{a}{\operatorname{argmax}}\, Q^{\pi_k}(s,a) \quad \forall s$

Asynchronous Policy Iteration

For $k = 1, 2, \ldots$

Pick any state $\hat{s}$

Calculate $Q^{\pi_k}(\hat{s}, a) \quad \forall a$

$\pi_{k+1}(\hat{s}) = \underset{a}{\operatorname{argmax}} \, Q^{\pi_k}(\hat{s}, a)$

and $\pi_{k+1}(s) = \pi_k(s) \quad \forall s \neq \hat{s}$

$Q^{\pi_{k+1}}(s,a) \geq Q^{\pi_k}(s,a) \quad \forall s,a$

$\mathbb{E}_{s\sim\rho}\left[V^{\pi_{k+1}}(s)\right] - \mathbb{E}_{s\sim\rho}\left[V^{\pi_k}(s)\right]$

$= \sum_{s,a} d^{\pi_{k+1}}(s)\left(\pi_{k+1}(a|s) - \pi_k(a|s)\right) Q^{\pi_k}(s,a)$

$= \sum_{a} d^{\pi_{k+1}}(\hat{s})\left(\pi_{k+1}(a|\hat{s}) - \pi_k(a|\hat{s})\right) Q^{\pi_k}(\hat{s},a)$

≥ 0

[If we want this to be positive & large]

Asynchronous Policy Iteration

- To improve policy, we may just evaluate Q^{π_k} on a particular state s.
- Of course, a **real improvement** is made only when $\exists a$ s.t. $Q^{\pi_k}(s,a) - V^{\pi_k}(s)$ is large.
- This is **different from Value Iteration**, where ideally, we would like to find Q_{k+1} such that $Q_{k+1}(s,a) \approx R(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)}\left[\max_{a'} Q_k(s',a')\right]$ $\forall s, a$
- VI-based algorithm like DQN usually requires **stronger function approximation** that can generalize to unseen state.

Policy Iteration with Samples

$s \to \boxed{} \to \pi(a|s)$

$Z_k(s_t,a) = \frac{r(s_t,a) - b(s_t)}{\pi_k(a|s_t)} \mathbb{1}(a_t = a)$

For $k = 1,\ 2, \ldots$ $\quad Z_k(s,a) = \hat{R}(s,a)$

For $i = 1, 2, \ldots, N$:

Choose action $a_i \sim \pi_{\theta_k}(\cdot\,|s_i)$ $\quad \mathbb{E}[Z_k(s,a)] \approx \hat{R}(s,a)$

Receive reward $r_i \sim R(s_i, a_i)$ and $s_i' \sim P(\cdot\,|s_i, a_i)$

$s_{i+1} = s_i'$ if episode continues, $s_{i+1} \sim \rho$ if episode ends

Data collection

$\approx R(s,a)$

Evaluate $\mathbb{E}[Z_k(s,a)] \approx Q^{\pi_{\theta_k}}(s,a)$ for $s = s_1, \ldots, s_N$ and all a

or $Z_k(s,a) \approx Q^{\pi_{\theta_k}}(s,a) - b_k(s)$ for $s = s_1, \ldots, s_N$ and all a

Policy Evaluation

Update θ_{k+1} from θ_k using the estimators $\{Z_k(s_i, a)\}_{i=1}^{N}$

Using any technique we introduced for policy-based contextual bandits

Policy Improvement

Why can we independently optimize the policy on each state?

Essentially treating **states** as **contexts**, but replacing $R(x, a)$ by $Q^{\pi_{\theta_k}}(s, a)$

Policy Evaluation

Policy Evaluation

Given: a policy π
Evaluate $V^{\pi}(s)$ or $Q^{\pi}(s,a)$

On-policy policy evaluation: the learner can execute π to evaluate π

Off-policy/offline policy evaluation: the learner can only execute some $\pi_b \neq \pi$, or can only access some existing dataset to evaluate π

Use cases:

- Approximate policy iteration: $\pi_k(s) = \operatorname*{argmax}_a Q^{\pi_{k-1}}(s,a)$
- Estimate the value of a policy before deploying it in the real world, e.g., COVID-related border measures, economic recovery policies, or policy changes in recommendation systems.

Value Iteration for V^π / Q^π

Assuming P, R are known

Input: π

For $k = 1,\ 2, \ldots$

$$\forall s, \quad V_k(s) \leftarrow \sum_a \pi(a|s)\left(R(s,a) + \gamma \sum_{s'} P(s'|s,a)\, V_{k-1}(s') \right)$$

Input: π

For $k = 1,\ 2, \ldots$

$Q_k \to Q^\pi$

$$\forall s, a, \quad Q_k(s,a) \leftarrow R(s,a) + \gamma \sum_{s',a'} P(s'|s,a)\, \pi(a'|s') Q_{k-1}(s',a')$$

On-Policy Policy Evaluation

Value Iterate for PE (given π)

Temporal Difference (TD) Learning for V^π

$\max_a$

correct π: collect

Q^{π^*}

For $k = 1, 2, \ldots$

Collect $\{(s_i, a_i, r_i, s_i')\}_{i=1}^N$ using policy π

$s_i' \sim P(\cdot | s_i, a_i)$

$$\theta_k \leftarrow \theta_{k-1} - \alpha \nabla_\theta \frac{1}{N} \sum_{i=1}^N \left(V_\theta(s_i) - r_i - \gamma V_{\theta_{k-1}}(s_i') \right)^2 \Bigg|_{\theta = \theta_{k-1}}$$

No target network needed because this is an **on-policy** problem.

This algorithm is also called TD(0).

Know P, R

$$V_k(s) \leftarrow \sum_a \pi(a|s) \left(R(s,a) + \gamma \sum_{s'} P(s'|s,a) V_{k-1}(s') \right)$$

$$= \sum_a \pi(a|s) R(s,a) + \gamma \sum_a \pi(a|s) \sum_{s'} P(s'|s,a) V_{k-1}(s')$$

with sample: collect (s_i, a_i, r_i, s_i') from π

$$\boxed{\min_\theta \sum_{i=1}^N \left(V_\theta(s_i) - (r_i + \gamma V_{\theta_{k-1}}(s_i')) \right)^2}$$

Temporal Difference (TD) Learning for Q^{π}

For $k = 1,\ 2, \ldots$

Collect $\{(s_i, a_i, r_i, s_i')\}_{i=1}^{N}$ using policy π

$$\theta_k \leftarrow \theta_{k-1} - \alpha \nabla_{\theta} \frac{1}{N} \sum_{i=1}^{N} \left(Q_{\theta}(s_i, a_i) - r_i - \gamma \sum_{a} \pi(a|s_i') Q_{\theta_{k-1}}(s_i', a') \right)^2 \Bigg|_{\theta=\theta_{k-1}}$$

No target network needed because this is an on-policy problem.

Monte Carlo Estimation

$Q^{\pi_{\theta_k}}(s,a) = \mathbb{E}\left[\sum_{h=1}^{\infty} \gamma^{h-1} R(s_h,a_h) \,\middle|\, (s_1,a_1)=(s,a)\right]$

Start from $(s_1, a_1) = (\hat{s}, \hat{a})$ and

$Q^{\pi}(\hat{s},\hat{a})$

execute policy π until the episode ends and obtain trajectory

$$s_1 = \hat{s}, a_1 = \hat{a}, r_1, s_2, a_2, r_2, \ldots, s_\tau, a_\tau, r_\tau$$

Let $G = \sum_{h=1}^{\tau} \gamma^{h-1} r_h$

$\mathbb{E}[G] = Q^{\pi}(\hat{s},\hat{a})$

$Z(\hat{s},a) = \dfrac{G\,\mathbb{I}\{a=\hat{a}\}}{\pi(\hat{a}|\hat{s})}$

G is an unbiased estimator for $Q^{\pi}(\hat{s}, \hat{a})$

band.t

r

$\mathbb{E}[r] = R(x_t, a_t)$

$\hat{R}(x_t, a_t)$

$r_1 + \gamma r_2 + \ldots$

$\hat{Q}(s,a)$

MC estimator: unbiased, higher variance

TD estimator: biased, lower variance

A Family of Estimators

TD(0) for V^{π}

Suppose we have a **state-value function estimation** $V_\phi(s) \approx V^{\pi}(s)$

Suppose we also have a **trajectory** $s_1, a_1, r_1, \ldots, s_\tau, a_\tau, r_\tau$ generated by π where $s_{\tau+1}$ is a terminal state

The following are all valid estimators of $Q^{\pi}(s_1, a_1)$:

$G_{1:1} = r_1 + \gamma V_\phi(s_2)$

$G_{1:2} = r_1 + \gamma r_2 + \gamma^2 V_\phi(s_3)$

…

$G_{1:\tau-1} = r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots + \gamma^{\tau-1} V_\phi(s_\tau)$

$G_{1:\tau} = r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots + \gamma^{\tau-1} r_\tau + \gamma^{\tau} V_\phi(s_{\tau+1})$

$G_{1:\tau+1} = r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots + \gamma^{\tau-1} r_\tau$

$G_{1:\tau+2}$

…

$\mathbb{E}[r_1 + \gamma V_\phi(s_2)] = R(s_1,a_1) + \sum_{s'} P(s'|s_1,a_1) V_\phi(s')$

$\approx R(s_1,a_1) + \sum_{s'} P(s'|s_1,a_1) V^{\pi}(s') = Q^{\pi}(s_1,a_1)$

$R(s_1,a_1) + \gamma R(s_2,a_2) + \gamma^2 \sum_{s'} P(s'|s,a) V_\phi(s')$

$\approx R(s_1,a_1) + \gamma R(s_2,a_2) + \gamma^2 \sum_{s'} P(s'|s,a) V^{\pi}(s')$

$\approx Q^{\pi}(s_1,a_1)$

A Family of Estimators

$\boxed{Q^{\pi}(s_1, a_1) - V^{\pi}(s_1)}$ = advange

And the following are estimators of $\boxed{Q^{\pi}(s_1, a_1) - V_{\phi}^{\pi}(s_1)}$ (baseline)

$A_{1:1} = r_1 + \gamma V_{\phi}(s_2) - V_{\phi}(s_1)$

...

$A_{1:\tau-1} = r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots + \gamma^{\tau-1} V_{\phi}(s_{\tau}) - V_{\phi}(s_1)$

$A_{1:\tau} = r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots + \gamma^{\tau-1} r_{\tau} - V_{\phi}(s_1)$

$A_{1:\tau+1} = r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots + \gamma^{\tau-1} r_{\tau} - V_{\phi}(s_1)$

...

Below, we will introduce a way to combine these estimators.

Balancing Bias and Variance

$G_{1:1}$, $G_{1:2}$, $\vdots$, $G_{1:\tau}$, $G_{1:\tau+1}$, $\vdots$, $G_{1:\infty}$ = estimators of $Q^{\pi}(s_1, a_1)$

same

$\lambda \in [0,1)$

$$G_1(\lambda) = (1-\lambda)\sum_{i=1}^{\infty} \lambda^{i-1} G_{1:i}$$

$$= (1-\lambda)\left(G_{1:1} + \lambda G_{1:2} + \lambda^2 G_{1:3} + \cdots + \lambda^{\tau-1} G_{1:\tau} + \lambda^{\tau} G_{1:\tau+1} + \lambda^{\tau+1} G_{1:\tau+2} + \cdots\right)$$

$$(1-\lambda)\left[1+\lambda+\lambda^2+\cdots\right] = 1$$

$$A_1(\lambda) = (1-\lambda)\sum_{i=1}^{\infty} \lambda^{i-1} A_{1:i}$$

$A_{1:i} = G_{1:i} - V_\phi(s_1)$

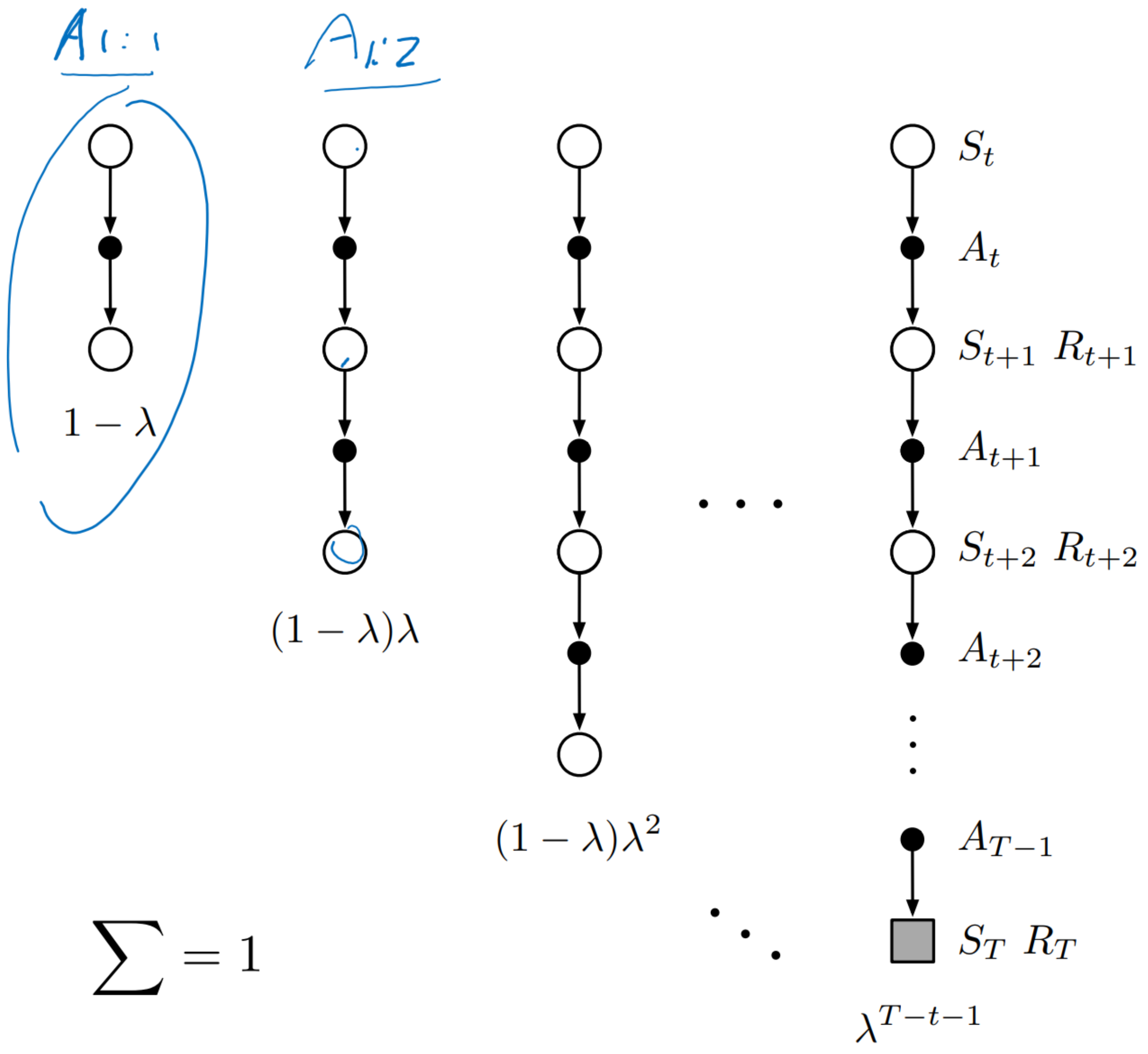
$\lambda \in [0,1)$

$$= (1-\lambda)\left(A_{1:1} + \lambda A_{1:2} + \lambda^2 A_{1:3} + \cdots + \lambda^{\tau-1} A_{1:\tau} + \lambda^{\tau} A_{1:\tau+1} + \lambda^{\tau+1} A_{1:\tau+2} + \cdots\right)$$

$A_1(\lambda) = G_1(\lambda) - V_\phi(s_1)$ **(Generalized Advantage Estimation)**

An estimation of $Q^{\pi}(s_1, a_1) - V^{\pi}(s_1)$

Balancing Bias and Variance



Computing Generalized Advantage Estimator (GAE)

We also need to calculate

$$A_2(\lambda) = (1-\lambda)\sum_{i=2}^{\infty} \lambda^{i-2} A_{2:i} \approx (1-\lambda)\left(A_{2:2} + \lambda A_{2:3} + \lambda^2 A_{2:4} + \cdots + \lambda^{\tau-2} A_{2:\tau} + \lambda^{\tau-1} A_{2:\tau+1} + \lambda^{\tau} A_{2:\tau+2} + \cdots\right)$$

same

$$A_3(\lambda) = (1-\lambda)\sum_{i=3}^{\infty} \lambda^{i-3} A_{3:i}$$

…

$Q^\pi(s_2,a_2) - V^\pi(s_2)$ . . .

$O(\tau)$

$A_1(\lambda), \cdots A_\tau(\lambda)$

To compute $A_1(\lambda), A_2(\lambda) \ldots, A_\tau(\lambda)$, naïve implementation takes $\tau \times \tau \times \tau$ time.

$Q^\pi(s_1,a_1) - V^\pi(s_1)$

Compute $A_{i:j}$

compute $A_i(\lambda)$

Efficient Computation of GAE (1/2)

$i \in [1, T]$

Define $\delta_i = r_i + \gamma V_\phi(s_{i+1}) - V_\phi(s_i) = A_{i:i}$

$$A_{i:j} = r_i + \gamma r_{i+1} + \gamma^2 r_{i+2} + \cdots + \gamma^{j-i} r_j + \gamma^{j-i+1} V_\phi(s_{j+1}) - V_\phi(s_i)$$

$$= r_i + \gamma r_{i+1} + \gamma^2 r_{i+2} + \cdots + \gamma^{j-i} r_j + \gamma^{j-i+1} V_\phi(s_{j+1}) - V_\phi(s_i)$$

$$= \delta_i + \gamma\delta_{i+1} + \gamma^2\delta_{i+1} + \cdots + \gamma^{j-i}\delta_j$$

$$= (r_i + \gamma V_\phi(s_{i+1}) - V_\phi(s_i)) + \gamma(r_{i+1} + \gamma V_\phi(s_{i+2}) - V_\phi(s_{i+1})) + \gamma^2(r_{i+2} + \gamma V_\phi(s_{i+3}) - V_\phi(s_{i+2}))$$

$$+ \cdots + \gamma^{j-i}(r_j + \gamma V_\phi(s_{j+1}) - V_\phi(s_j))$$

Efficient Computation of GAE (2/2)

$$A_\tau(\lambda) = (1-\lambda)\left(A_{\tau:\tau} + \lambda A_{\tau:\tau+1} + \lambda^2 A_{\tau:\tau+2} + \cdots\right) = A_{\tau\tau} = \delta_\tau = r_\tau + \cancel{\gamma V_\phi(s_{\tau+1})}^{\,0} - V_\phi(s_\tau)$$

$$A_{\tau-1}(\lambda) = (1-\lambda)\left(A_{\tau-1:\tau-1} + \lambda A_{\tau-1:\tau} + \lambda^2 A_{\tau-1:\tau+1} + \cdots\right)$$

$$\vdots$$

$$A_1(\lambda) = (1-\lambda)\left(A_{1:1} + \lambda A_{1:2} + \lambda^2 A_{1:3} + \cdots\right)$$

$$A_{i:j} = \delta_i + \gamma\delta_{i+1} + \gamma^2\delta_{i+2} + \cdots + \gamma^{j-i}\delta_j$$

$$\boxed{A_i(\lambda) = \delta_i + \lambda\gamma A_{i+1}(\lambda)}$$

$$A_i(\lambda) = (1-\lambda)\left[A_{i:i} + \lambda A_{i:i+1} + \lambda^2 A_{i:i+2} + \lambda^3 A_{i:i+3} + \cdots\right]$$

$$= (1-\lambda)\left[\delta_i + \lambda(\delta_i + \gamma\delta_{i+1}) + \lambda^2(\delta_i + \gamma\delta_{i+1} + \gamma^2\delta_{i+2}) + \lambda^3(\delta_i + \gamma\delta_{i+1} + \gamma^2\delta_{i+2} + \gamma^3\delta_{i+3}) + \cdots\right]$$

$$= \boxed{(1-\lambda)}\left[\underline{(1+\lambda+\lambda^2+\lambda^3+\cdots)}\,\delta_i + \lambda\gamma\boxed{\left(\delta_{i+1} + \lambda(\delta_{i+1} + \gamma\delta_{i+2}) + \lambda^2(\delta_{i+1} + \gamma\delta_{i+2} + \gamma^2\delta_{i+3}) + \cdots\right)}\right]$$

$$= \delta_i + \lambda\gamma A_{i+1}(\lambda)$$

GAE (Generalized Advantage Estimation)

Let $(s_1, a_1, r_1, s_1', s_2, a_2, r_2, s_2', \dots, s_N, a_N, r_N, s_N')$ be a trajectory collected with policy π, where $s_i' = s_{i+1}$ if s_i' is not a terminal state, and $s_{i+1} \sim \rho$ otherwise.

Also, let V_ϕ be a given state-value estimation.

Then the following procedure can estimate $A_i \approx \boxed{Q^\pi(s_i, a_i) - V_\phi(s_i)}$ $i \in \{1, \dots, N\}$

Parameter: λ (controlling variance-bias tradeoff)

For $i = N, N-1, \dots, 1$:

If s_i' is a terminal state:

$$\delta_i = r_i - V_\phi(s_i)$$

$$A_i = \delta_i$$

Else:

$$\delta_i = r_i + \gamma V_\phi(s_{i+1}) - V_\phi(s_i)$$

$$A_i = \delta_i + \lambda\gamma A_{i+1}$$

Schulman et al. High-Dimensional Continuous Control Using Generalized Advantage Estimation. 2015.

Using GAE in the Policy Iteration Framework

For $k = 1,\ 2, \ldots$

For $i = 1, 2, \ldots, N$:

Choose action $a_i \sim \pi_{\theta_k}(\cdot\,|s_i)$

Receive reward $r_i \sim R(s_i, a_i)$ and $s_i' \sim P(\cdot\,|s_i, a_i)$

$s_{i+1} = s_i'$ if episode continues, $s_{i+1} \sim \rho$ if episode ends

Data collection

$V^{\pi_{\theta_k}}(s)$

Evaluate $Z_k(s,a) \approx Q^{\pi_{\theta_k}}(s,a) - V_\phi(s)$ for $s = s_1, \ldots, s_N$ and all a

$\Rightarrow\ Z_k(s_i, a) = \dfrac{\mathbb{I}\{a_i = a\}}{\pi_{\theta_k}(a|s_i)} \hat{A}_k(s_i, a_i)$

$\boxed{A_\pi(x)}$ in the previous side

Policy Evaluation

Update θ_{k+1} from θ_k using the estimator $\{Z_k(s_i, a)\}_{i=1}^{N}$

Using any technique we introduced for policy-based contextual bandits

Policy Improvement

Training the Baseline V_ϕ (in iteration k)

For $k = 1, 2, \ldots$

$V_\phi(s) \approx V^{\pi_{\theta_k}}(s)$

$V^\pi(s)$

For $i = 1, 2, \ldots, N$:

Choose action $a_i \sim \pi_{\theta_k}(\cdot \mid s_i)$

Receive reward $r_i \sim R(s_i, a_i)$ and $s_i' \sim P(\cdot \mid s_i, a_i)$

$s_{i+1} = s_i'$ if episode continues, $s_{i+1} \sim \rho$ if episode ends

$r_i + \gamma V_{\phi_k}(s_{i+1})$

$r_i + \gamma r_{i+1} + \gamma^2 V_{\phi_k}(s_{i+2})$

$\vdots$

$$\phi_{k+1} \leftarrow \phi_k - \alpha \nabla_\phi \frac{1}{N} \sum_{i=1}^{N} \left(V_\phi(s_i) - r_i - \gamma V_{\phi_k}(s_i') \right)^2 \Bigg|_{\phi=\phi_k} \qquad \text{TD(0)}$$

$$\phi_{k+1} \leftarrow \phi_k - \alpha \nabla_\phi \frac{1}{N} \sum_{i=1}^{N} \left(V_\phi(s_i) - G_i(\lambda; \phi_k) \right)^2 \Bigg|_{\phi=\phi_k} \quad \text{where } G_i(\lambda; \phi_k) = A_i(\lambda; \phi_k) + V_{\phi_k}(s_i) \qquad \text{TD}(\lambda)$$

$$\phi_{k+1} \leftarrow \phi_k - \alpha \nabla_\phi \frac{1}{N} \sum_{i=1}^{N} \left(V_\phi(s_i) - \sum_{h=i}^{\tau(i)} \gamma^{h-i} r_i \right)^2 \Bigg|_{\phi=\phi_k} \qquad \text{TD(1)}$$

Approximate Policy Iteration and Variants

PPO

GAE $A_i(\lambda)$

Requires training a separate V_ϕ

For $k = 1, 2, \ldots$

For $i = 1, 2, \ldots, N$:

Choose action $a_i \sim \pi_{\theta_k}(\cdot\,|s_i)$

Receive reward $r_i \sim R(s_i, a_i)$ and $s_i' \sim P(\cdot\,|s_i, a_i)$

$s_{i+1} = s_i'$ if episode continues, $s_{i+1} \sim \rho$ if episode ends

Define $Z_k(s_i, a) = \frac{\mathbb{I}\{a_i = a\}}{\pi_{\theta_k}(a|s_i)} \boxed{\hat{A}_k(s_i, a_i)}$

Use another inner for-loop to solve the argmax with gradient ascent

$$\theta_{k+1} = \operatorname*{argmax}_\theta \left\{ \frac{1}{N} \sum_{i=1}^{N} \left(\sum_a \pi_\theta(a|s_i) Z_k(s_i, a) - \frac{1}{\eta} \mathrm{KL}\big(\pi_{\theta_k}(\cdot\,|s_i), \pi_\theta(\cdot\,|s_i)\big) \right) \right\}$$

$$\approx \operatorname*{argmax}_\theta \left\{ \frac{1}{N} \sum_{i=1}^{N} \left(\frac{\pi_\theta(a_i|s_i)}{\pi_{\theta_k}(a_i|s_i)} \hat{A}_k(s_i, a_i) - \frac{1}{\eta} \left(\frac{\pi_\theta(a_i|s_i)}{\pi_{\theta_k}(a_i|s_i)} - 1 - \log \frac{\pi_\theta(a_i|s_i)}{\pi_{\theta_k}(a_i|s_i)} \right) \right) \right\}$$

Schulman et al. Proximal Policy Optimization Algorithms. 2017.

PPO with Clipping

Schulman et al. Proximal Policy Optimization Algorithms. 2017.

① Clipping ② KL-regularization

$$\theta_{k+1} = \underset{\theta}{\operatorname{argmax}} \left\{ \frac{1}{N} \sum_{i=1}^{N} \left(\boxed{} \cancel{- \frac{1}{\eta} \left(\frac{\pi_\theta(a_i|s_i)}{\pi_{\theta_k}(a_i|s_i)} - 1 - \log \frac{\pi_\theta(a_i|s_i)}{\pi_{\theta_k}(a_i|s_i)} \right)} \right) \right\}$$

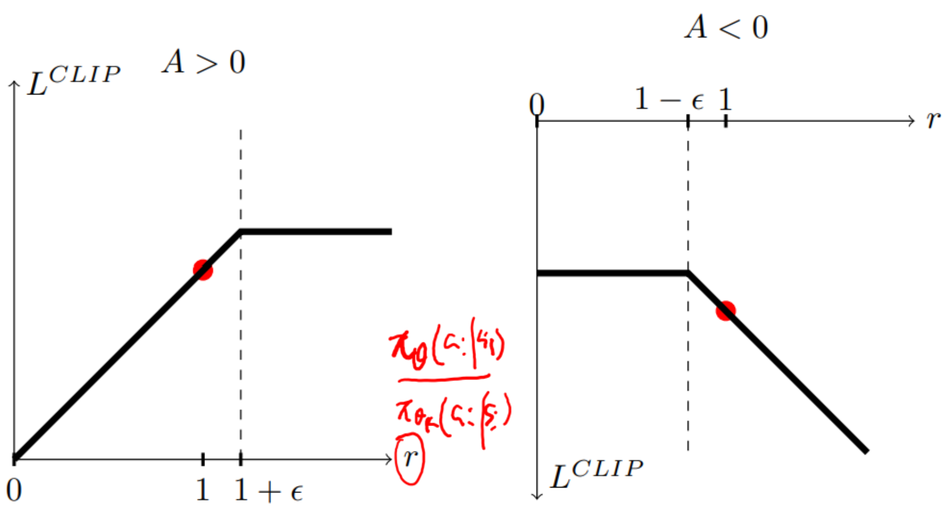
$< 1-\epsilon$

$\epsilon \approx 0.1, 0.2$

$$L_{CLIP} \triangleq \min \left\{ \frac{\pi_\theta(a_i|s_i)}{\pi_{\theta_k}(a_i|s_i)} \hat{A}_k(s_i,a_i), \quad \text{clip}_{[1-\epsilon,1+\epsilon]} \left(\frac{\pi_\theta(a_i|s_i)}{\pi_{\theta_k}(a_i|s_i)} \right) \hat{A}_k(s_i,a_i) \right\}$$

smaller

$1-\epsilon$



Preventing $\frac{\pi_{\theta_{k+1}}(a_i|s_i)}{\pi_{\theta_k}(a_i|s_i)} \hat{A}_k(s_i,a_i)$ from being too high

~~from being too low~~

A2C (Advantage Actor Critic) / PG

For $k = 1,\ 2, \ldots$

For $i = 1, 2, \ldots, N$:

Choose action $a_i \sim \pi_{\theta_k}(\cdot\,|s_i)$

Receive reward $r_i \sim R(s_i, a_i)$ and $s_i' \sim P(\cdot\,|s_i, a_i)$

$s_{i+1} = s_i'$ if episode continues, $s_{i+1} \sim \rho$ if episode ends

$$\theta_{k+1} = \theta_k - \eta \frac{1}{N} \sum_{i=1}^{N} \left(\nabla_\theta \log \pi_\theta(a_i|s_i)\right)\Big|_{\theta=\theta_k} \hat{A}_k(s_i, a_i)$$

In standard A2C, $\hat{A}_k(s_i, a_i) = \underline{r_i + \gamma V_{\phi_k}(s_i') - V_{\phi_k}(s_i)}$ (GAE estimator with $\lambda = 0$)

and ϕ_k is trained with TD(0):

$$\phi_{k+1} \leftarrow \phi_k - \alpha \nabla_\phi \frac{1}{N} \sum_{i=1}^{N} \left(V_\phi(s_i) - r_1 - \gamma V_{\phi_k}(s_i')\right)^2\Bigg|_{\phi=\phi_k}$$

A2C (Advantage Actor Critic) / PG

For $k = 1,\ 2, \ldots$

For $i = 1, 2, \ldots, N$:

Choose action $a_i \sim \pi_{\theta_k}(\cdot\,|s_i)$

Receive reward $r_i \sim R(s_i, a_i)$ and $s_i' \sim P(\cdot\,|s_i, a_i)$

$s_{i+1} = s_i'$ if episode continues, $s_{i+1} \sim \rho$ if episode ends

$$\theta_{k+1} = \theta_k - \eta \frac{1}{N} \sum_{i=1}^{N} \left(\nabla_\theta \log \pi_\theta(a_i|s_i)\right)\Big|_{\theta=\theta_k} \hat{A}_k(s_i, a_i)$$

In standard PG, $\hat{A}_k(s_i, a_i) = \boxed{\sum_{h=i}^{\tau(i)} \gamma^{h-i} r_i} - \underline{V_{\phi_k}(s_i)}$ (GAE estimator with $\underline{\lambda = 1}$)

A2C (Advantage Actor Critic) / PG

For $k = 1,\ 2, \ldots$

For $i = 1, 2, \ldots, N$:

Choose action $a_i \sim \pi_{\theta_k}(\cdot\,|s_i)$

Receive reward $r_i \sim R(s_i, a_i)$ and $s_i' \sim P(\cdot\,|s_i, a_i)$

$s_{i+1} = s_i'$ if episode continues, $s_{i+1} \sim \rho$ if episode ends

$$\theta_{k+1} = \theta_k - \eta \frac{1}{N} \sum_{i=1}^{N} \left(\nabla_\theta \log \pi_\theta(a_i|s_i)\right)\Big|_{\theta=\theta_k} \hat{A}_k(s_i, a_i)$$

$\underline{V_\phi,\ \lambda}$

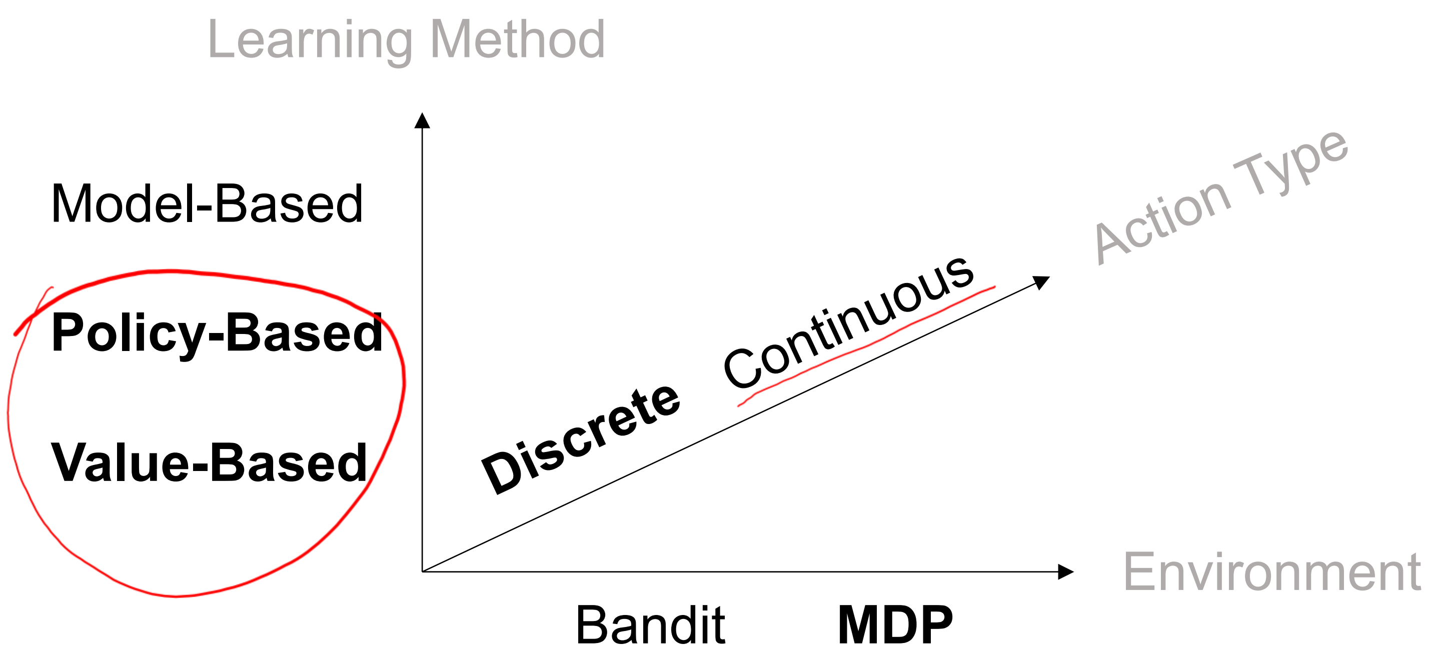
In general, one can use GAE with any λ to calculate $\hat{A}_k(s_i, a_i)$, with V_ϕ calculated from TD(λ') with any λ'.

Summary: Algorithms based on Policy Iteration

- The algorithms are almost the same as those we introduced for contextual bandits
  - PPO / NPG ← KL regularized / Clipped
  - A2C / PG ←
- The only change is replacing $r(x_i, a_i) - b(x_i)$ by Advantage Estimator:
  - $\lambda = 0$: $r(s_i, a_i) + \gamma V_\phi(s_{i+1}) - V_\phi(s_i)$
  - $\lambda = 1$: $r(s_i, a_i) + \gamma r(s_{i+1}, a_{i+1}) + \gamma^2 r(s_{i+2}, a_{i+2}) + \cdots + \gamma^{\tau - i} r(s_\tau, a_\tau) - V_\phi(s_i)$
  - Any $\lambda \in [0,1]$: calculated by the GAE procedure
- The baseline $V_\phi(s)$ tries to track $V^{\pi_\theta}(s)$ where π_θ is the current policy
  - It is trained with a separate procedure TD(λ')

$$\phi_{k+1} \leftarrow \phi_k - \alpha \nabla_\phi \frac{1}{N} \sum_{i=1}^{N} \left(V_{\phi}(s_i) - r_1 - \gamma V_{\phi_k}(s_i') \right)^2 \Bigg|_{\phi = \phi_k} \qquad \text{TD(0)}$$

Roadmap


Learning Method
Model-Based
Policy-Based
Value-Based
Action Type
Discrete
Continuous
Environment
Bandit
MDP