

Approximate Policy Iteration and Variants

Chen-Yu Wei

Policy Iteration

For $k = 1, 2, \dots$

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

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

Asynchronous Policy Iteration

For $k = 1, 2, \dots$

Pick any state \hat{s}

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

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

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

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

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

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

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

$$\geq 0$$

[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 $Q_{k+1}(s, a) \approx R(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s, a)} \left[\max_{a'} Q_k(s', a') \right] \quad \forall s, a$
- VI-based algorithm like DQN usually requires **stronger function approximation** that can generalize to unseen state.

Policy Iteration with Samples

$$s \rightarrow \boxed{\cdot} \rightarrow \pi(a|s)$$

For $k = 1, 2, \dots$

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

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

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

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

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

Data collection

Policy Evaluation

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

Input: π

For $k = 1, 2, \dots$

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

Input: π

For $k = 1, 2, \dots$

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

On-Policy Policy Evaluation

Temporal Difference (TD) Learning for V^π

For $k = 1, 2, \dots$

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

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

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

This algorithm is also called TD(0)

Temporal Difference (TD) Learning for Q^π

For $k = 1, 2, \dots$

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

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

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

Monte Carlo Estimation

Start from $(s_1, a_1) = (\hat{s}, \hat{a})$ and execute policy π until the episode ends and obtain trajectory

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

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

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

MC estimator: unbiased, higher variance

TD estimator: biased, lower variance

A Family of Estimators

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

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

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

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

...

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

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

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

...

A Family of Estimators

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

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

...

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

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

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

...

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

Balancing Bias and Variance

$$\begin{aligned} G_1(\lambda) &= (1 - \lambda) \sum_{i=1}^{\infty} \lambda^{i-1} G_{1:i} \\ &= (1 - \lambda) (G_{1:1} + \lambda G_{1:2} + \lambda^2 G_{1:3} + \cdots + \lambda^{\tau-1} G_{1:\tau} + \lambda^{\tau} G_{1:\tau+1} + \lambda^{\tau+1} G_{1:\tau+2} + \cdots) \end{aligned}$$

$$\begin{aligned} A_1(\lambda) &= (1 - \lambda) \sum_{i=1}^{\infty} \lambda^{i-1} A_{1:i} && \textbf{(Generalized Advantage Estimation)} \\ &= (1 - \lambda) (A_{1:1} + \lambda A_{1:2} + \lambda^2 A_{1:3} + \cdots + \lambda^{\tau-1} A_{1:\tau} + \lambda^{\tau} A_{1:\tau+1} + \lambda^{\tau+1} A_{1:\tau+2} + \cdots) \end{aligned}$$

$$A_1(\lambda) = G_1(\lambda) - V_{\phi}(s_1)$$

Computing Generalized Advantage Estimator (GAE)

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 $A_i \approx Q^\pi(s_i, a_i) - V_\phi(s_i)$

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:

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

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

Using GAE in the Policy Iteration Framework

For $k = 1, 2, \dots$

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

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

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

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

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

Data collection

Policy Evaluation

Policy Improvement

Training the Baseline V_ϕ (in iteration k)

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

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

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

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

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

Approximate Policy Iteration and Variants

PPO

For $k = 1, 2, \dots$

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

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

Requires training a separate V_ϕ

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

$$\begin{aligned} \theta_{k+1} &= \operatorname{argmax}_{\theta} \left\{ \frac{1}{N} \sum_{i=1}^N \left(\sum_a \pi_{\theta}(a|s_i) Z_k(s_i, a) - \frac{1}{\eta} \operatorname{KL}(\pi_{\theta_k}(\cdot | s_i), \pi_{\theta}(\cdot | s_i)) \right) \right\} \\ &\approx \operatorname{argmax}_{\theta} \left\{ \frac{1}{N} \sum_{i=1}^N \left(\frac{\pi_{\theta}(a_i|s_i)}{\pi_{\theta_k}(a_i|s_i)} \hat{A}_k(s_i, a_i) - \frac{1}{\eta} \left(\frac{\pi_{\theta}(a_i|s_i)}{\pi_{\theta_k}(a_i|s_i)} - 1 - \log \frac{\pi_{\theta}(a_i|s_i)}{\pi_{\theta_k}(a_i|s_i)} \right) \right) \right\} \end{aligned}$$

PPO with Clipping

$$\theta_{k+1} = \operatorname{argmax}_{\theta} \left\{ \frac{1}{N} \sum_{i=1}^N \left(\boxed{\phantom{\min \left\{ \frac{\pi_{\theta}(a_i|s_i)}{\pi_{\theta_k}(a_i|s_i)} \hat{A}_k(s_i, a_i), \operatorname{clip}_{[1-\epsilon, 1+\epsilon]} \left(\frac{\pi_{\theta}(a_i|s_i)}{\pi_{\theta_k}(a_i|s_i)} \right) \hat{A}_k(s_i, a_i) \right\}}} - \frac{1}{\eta} \left(\frac{\pi_{\theta}(a_i|s_i)}{\pi_{\theta_k}(a_i|s_i)} - 1 - \log \frac{\pi_{\theta}(a_i|s_i)}{\pi_{\theta_k}(a_i|s_i)} \right) \right) \right\}$$

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

A2C (Advantage Actor Critic) / PG

For $k = 1, 2, \dots$

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

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, $\hat{A}_k(s_i, a_i) = r_i + \gamma V_{\phi_k}(s'_i) - V_{\phi_k}(s_i)$ (GAE estimator with $\lambda = 0$)
and ϕ_k is trained with TD(0):

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

A2C (Advantage Actor Critic) / PG

For $k = 1, 2, \dots$

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

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, $\hat{A}_k(s_i, a_i) = \sum_{h=i}^{\tau(i)} \gamma^{h-i} r_h - V_{\phi_k}(s_i)$ (GAE estimator with $\lambda = 1$)

A2C (Advantage Actor Critic) / PG

For $k = 1, 2, \dots$

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

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 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
 - A2C / PG
- The only change is replacing $r(x_i, a_i) - b(x_i)$ by Advantage Estimator:
 - $\lambda = 0$: $r(s_i, a_i) + \gamma V_\phi(s_{i+1}) - V_\phi(s_i)$
 - $\lambda = 1$: $r(s_i, a_i) + \gamma r(s_{i+1}, a_{i+1}) + \gamma^2 r(s_{i+2}, a_{i+2}) + \dots + \gamma^{\tau-i} r(s_\tau, a_\tau) - V_\phi(s_i)$
 - 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 $\text{TD}(\lambda')$

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