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

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

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

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

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

Evaluate $\hat{Q}_k(s,a) \approx\ Q^{\pi_{\theta_k}}(s,a)$ for $s = s_1, \ldots, s_N$ and all a

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

Policy Evaluation

Update θ_{k+1} from θ_k using $\hat{Q}_k(s,a)$ or $\hat{A}_k(s,a)$

Using any technique we introduced for policy-based contextual bandits

Just replacing $r(x,a) - b(x)$ by $\hat{Q}_k(s,a)$ or $\hat{A}_k(s,a)$

Policy Improvement

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: $\pi_k(s) = \underset{a}{\operatorname{argmax}}\, Q^{\pi_{k-1}}(s, a)$
- 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 $k = 1,\ 2, \ldots$

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

Input: π

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

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

On-Policy Policy Evaluation

Temporal Difference (TD) Learning for V^π

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

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

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

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

This algorithm is also called TD(0)

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

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** $V_\theta(s) \approx V^\pi(s)$

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 $Q^\pi(s_1, a_1)$:

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

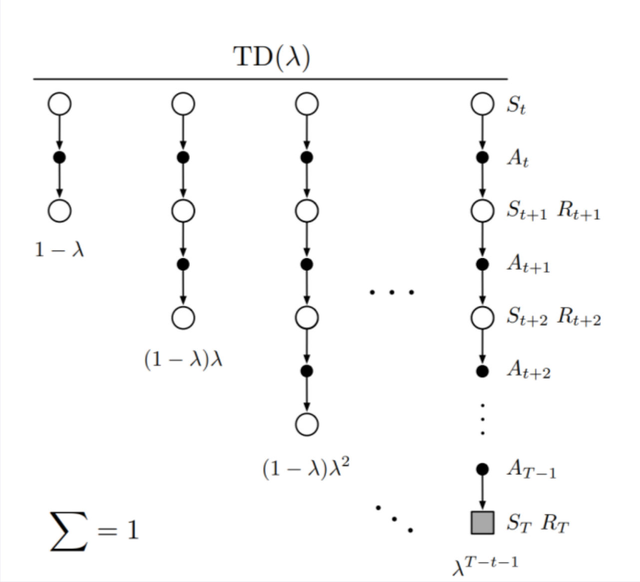
...

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

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

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

…



A Family of Estimators

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

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

...

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

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

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

…

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

Balancing Bias and Variance

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

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

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

(Generalized Advantage Estimation)

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

$$A_1(\lambda) = G_1(\lambda) - V_\theta(s_1)$$

Computing Generalized Advantage Estimator (GAE)

Using GAE in 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

Train $V_\phi(s)$ using Temporal Difference Learning

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

Policy Evaluation

Update θ_{k+1} from θ_k using $\hat{A}_k(s, a)$

Using any technique we introduced for policy-based contextual bandits

Just replacing $r(x, a) - b(x)$ by $\hat{A}_k(s, a)$

Policy Improvement

TD(λ)

TD(0): $\theta_{k+1} \leftarrow \theta_k - \alpha \nabla_{\theta} \left(V_{\theta}(s_1) - r_1 - \gamma V_{\theta_k}(s_2) \right)^2 \Big|_{\theta=\theta_k}$

TD(λ): $\theta_{k+1} \leftarrow \theta_k - \alpha \nabla_{\theta} \left(V_{\theta}(s_1) - G_1(\lambda) \right)^2 \Big|_{\theta=\theta_k}$