Off-Policy Algorithm

1. Preliminaries

- Every MDP is a set of Markov chain, which we denote $MDP(\mathcal{S}, \mathcal{A}, p_0, p(s'|s,a), r(s,a,s'))$;
  - We denote $\mathcal{S} = \{s1, s2, \dots sn\}$;
  - $P_\pi(s,s') = \sum_a \pi(a|s)p(s'|s,a)$;
  - $r_\pi(s) = \sum_a \pi(a|s) \sum_{s'} p(s'|s,a)r(s,a,s')$.
- In other words, we can see MDP as a function mapping a policy $\pi \in \Pi$ to a Markov chain: $MDP : \Pi \to MC$, or $MDP = \{\pi \to MC_\pi\}$ where $MC_\pi = \{\tau = (s_0, a_0, r_0, s_t \dots)|s_0 \sim p_0, a_t \sim \pi(\cdot|s_t), s_{t+1} \sim p_\pi(\cdot|s_t, a_t), r_t \sim r_\pi(s_t, a_t, s_{t+1})\}$;
  - We denote MC_π's stationary distribution is d_π and $D_\pi = diag(d_\pi)$;
  - State-transition space: $MC2_\pi = \{(s,a,r,s')|s \sim d_\pi, a \sim \pi(\cdot|s), r \sim r(s,a,s'), s' \sim p(\cdot|s,a)\}$;
  - The key assumption MC_π can break into $MC2_\pi$.
- Criterion:
  - $R(\tau) = \sum_{t=0}^{T} \gamma^t r_t$;
  - State value function: $V_\pi(s) = \mathbb{E}[R(\tau)|\tau \in MC_\pi, \tau(s_0) = s]$;
  - State-action value function: $Q_\pi(s,a) = \mathbb{E}[R(\tau)|\tau \in MC_\pi, \tau(s_0) = s, \tau(a_0) = a]$.
- Off-policy settings: $data_\mu \sim MC_\mu$.

2. TD Algorithm(Policy Evaluation, Critic)

2.1 On policy TD Algorithm

- The TD(0) algorithm is $V(s_t) = V(s_t) + \alpha_t(r_t + \gamma V(s_{t+1}) - V(s_t))$;

  If we use linear function to approximate $\tilde{V}_\theta = \Phi\theta \approx V^\pi$, where

  $$\Phi = [\phi(s1), \phi(s2), \dots, \phi(sn)]^T,$$

  then the TD(0) algorithm becomes

  $$\begin{aligned}\theta =& \theta + \alpha(r_t + \gamma\tilde{V}_\theta(s_{t+1}) - \tilde{V}_\theta(s_t))\nabla_\theta\tilde{V}_\theta(s_t)\\ =& \theta + \alpha(r_t + \gamma\tilde{V}_\theta(s_{t+1}) - \tilde{V}_\theta(s_t))\phi(s_t)\end{aligned}$$

- The TD(0) algorithm with linear function approximation converges to $\tilde{V}_{\theta^*} = \Pi_\pi T_\pi \tilde{V}_{\theta^*}$; (Tsitsiklis 1998)
  - Linear projection $\Pi_\pi = \Phi(\Phi^T D_\pi \Phi)^{-1}\Phi^T D_\pi$;

    (Hint: From problem $\arg\min_\theta \|\Phi\theta - V\|_{d_\pi}$, we can get $\theta^* = (\Phi^T D_\mu \Phi)^{-1}\Phi^T D_\mu V$.)
  - Bellman projection $T_\pi V = r_\pi + \gamma P_\pi V$;
  - $\|\tilde{V}_{\theta^*} - V^\pi\|_{D_\pi} = \|\Phi\theta^* - V^\pi\|_{D_\pi} \le \frac{1}{1-\gamma}\|\Pi_\pi V^\pi - V^\pi\|_{D_\pi}$.

2.2 Off-policy TD Algorithm

2.2.1 Monte carlo method in trajectory space

- Important sampling method in trajectory space:
  - $$MC_\pi = \{\tau = (s_0, a_0, r_0, \dots)|s_0 \sim p_0, a_t \sim \pi(\cdot|s_t), s_{t+1} \sim p(\cdot|s_t, a_t), r_t \sim r(s_t, a_t, s_{t+1})\}$$
    ;
  - $$MC_\mu = \{\tau = (s_0, a_0, r_0, \dots)|s_0 \sim p_0, a_t \sim \mu(\cdot|s_t), s_{t+1} \sim p_\mu(\cdot|s_t, a_t), r_t \sim r(s_t, a_t, s_{t+1})\}$$
    .
  - $P(\tau_\pi) = p_0(s_0)\pi(a_0|s_0)p(s_1|s_0, a_0) \cdots \pi(a_t|s_t)p(s_{t+1}|s_t, a_t) \cdots$;
  - $P(\tau_\mu) = p_0(s_0)\mu(a_0|s_0)p(s_1|s_0, a_0) \cdots \mu(a_t|s_t)p(s_{t+1}|s_t, a_t) \cdots$.
- $V(s_t) = V(s_t) + \alpha\rho_t(r_t + \gamma V(s_{t+1}) - V(s_t))$;
  - $\rho_t = \frac{P(\tau_\pi)}{P(\tau_\mu)}$;
  - $\rho_t = \frac{\pi(a_0|s_0)}{\mu(a_0|s_0)} \cdot \frac{\pi(a_1|s_1)}{\mu(a_1|s_1)} \cdots \frac{\pi(a_t|s_t)}{\mu(a_t|s_t)}$.
- The problem of Monte carlo methods:
  - ρ_t can easily be zero;
  - high variance.

2.2.2 TD method in state-transition space

- Intuitively, we are only interested in the fixed point of bellman operator $V = T^\pi V$, which is exactly V^π, no matter what norm the algorithm uses.
- Important sampling method in state-transition space:
  - $MC2_\pi = \{(s, a, r, s')|s \sim d_\pi, a \sim \pi(\cdot|s), r \sim r(s, a, s'), s' \sim p(\cdot|s, a)\}$;
  - $MC2_\mu = \{(s, a, r, s')|s \sim d_\mu, a \sim \mu(\cdot|s), r \sim r(s, a, s'), s' \sim p(\cdot|s, a)\}$;
  - $MC2_{\pi,\mu} = \{(s, a, r, s')|s \sim d_\mu, a \sim \pi(\cdot|s), r \sim r(s, a, s'), s' \sim p(\cdot|s, a)\}$.
- $V(s_t) = V(s_t) + \alpha\rho_t(r_t + \gamma V(s_{t+1}) - V(s_t))$;
  - Correct samples from $MC2_\mu$ to $MC2_{\pi,\mu}$: $\rho_t = \frac{\pi(a_t|s_t)}{\mu(a_t|s_t)}$;(Old method: GTD, GTD2, TDC)
  - Correct samples from $MC2_\mu$ to $MC2_\pi$: $\rho_t = \frac{d_\pi(s_t)}{d_\mu(s_t)} \cdot \frac{\pi(a_t|s_t)}{\mu(a_t|s_t)}$. (New method: 2018~2019)

2.2.3 Vanilla Off-policy TD Algorithm

- $V(s_t) = V(s_t) + \alpha_t\rho_t(r_t + \gamma V(s_{t+1}) - V(s_t))$
- Unstable example:

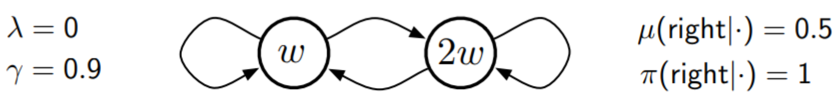

Figure 1: $w \to 2w$ example without a terminal state.

- A good conclusion of TD-algorithm:

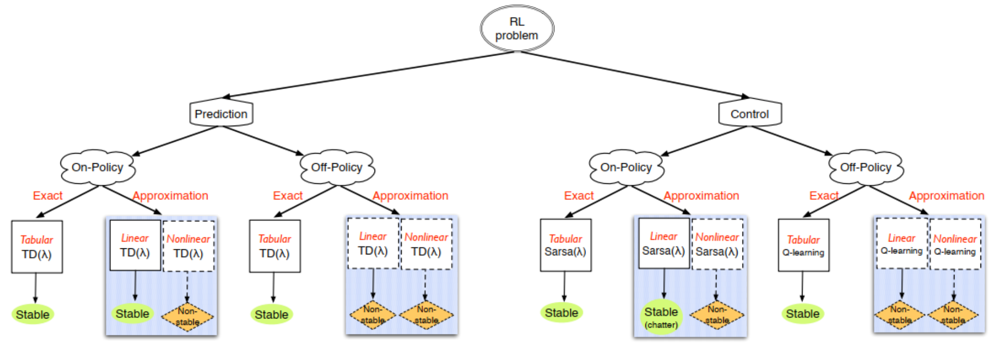

Figure 1.1: Status of conventional TD methods with tabular representation and function approximation, in terms of stability. For stability analysis of linear Sarsa (λ) as well as its chattering behavior see Gordon (2000).

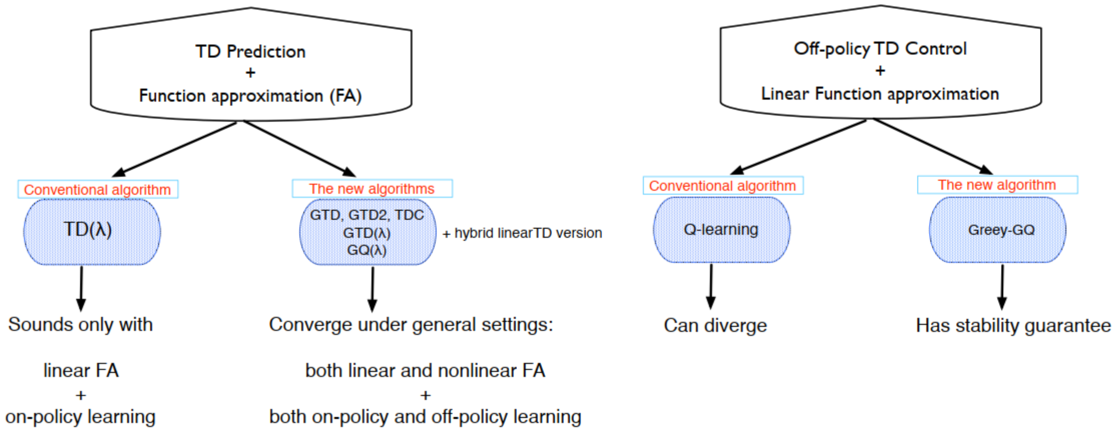

Figure 1.2: Algorithmic contributions.

2.2.4 Gradient TD Algorithm

Let $\delta(s,a,r,s') = (r + \gamma\phi^T(s')\theta - \phi^T(s)\theta)$.

- GTD algorithm:
  - **Objective**: The norm of the expected TD update $NEU(\theta) = ||\mathbb{E}_{(s,a,r,s')\sim MC2_{\pi,\mu}}[\delta(s,a,r,s')\phi(s)]||_2^2$;
  - The deviation of objective is

$$
\begin{aligned}
-\frac{1}{2}\nabla_\theta NEU(\theta) =& \mathbb{E}_{(s,a,r,s')\sim MC2_{\pi,\mu}}[(\phi(s) - \gamma\phi(s'))\phi^T(s)] \\
&\cdot \mathbb{E}_{(s,a,r,s')\sim MC2_{\pi,\mu}}[\delta(s,a,r,s')\phi(s)];
\end{aligned}
$$

  - Algorithm step:
    - $\theta_{t+1} = \theta_t + \alpha_t\rho_t(\phi(s_t) - \gamma\phi(s_{t+1}))\phi^T(s_t)w_t$;
    - $w_{t+1} = w_t + \beta_t(\rho_t\delta(s_t,a_t,r_t,s_{t+1})\phi(s_t) - w_t)$.
- GTD2 algorithm:
  - **Objective**: $J(\theta) = ||V_\theta - \Pi_\mu T^\pi V_\theta||_{d_\mu}^2$;

$$
\begin{aligned}
J(\theta) = & \mathbb{E}_{(s,a,r,s')\sim MC2_{\pi,\mu}}[\delta(s,a,r,s')\phi(s)]^T \\
& \cdot (\mathbb{E}_{(s,a,r,s')\sim MC2_{\pi,\mu}}[\phi(s)\phi^T(s)])^{-1} \\
& \cdot \mathbb{E}_{(s,a,r,s')\sim MC2_{\pi,\mu}}[\delta(s,a,r,s')\phi(s)]
\end{aligned}
$$

  - The deviation of objective is

$$-\frac{1}{2}\nabla J(\theta) = \mathbb{E}_{(s,a,r,s')\sim MC2_{\pi,\mu}}[(\phi(s) - \gamma\phi(s'))\phi^T(s)] \\ \cdot (\mathbb{E}_{s\sim d_\mu}[\phi(s)\phi^T(s)])^{-1} \\ \cdot \mathbb{E}_{(s,a,r,s')\sim MC2_{\pi,\mu}}[\delta(s,a,r,s')\phi(s)]$$

- Algorithm step:
  - $\theta_{t+1} = \theta_t + \alpha_t\rho_t(\phi(s_t) - \gamma\phi(s_{t+1}))\phi^T(s_t)w_t$;
  - $w_{t+1} = w_t + \beta_t(\rho_t\delta(s_t, a_t, r_t, s_{t+1}) - \phi^T(s_t)w_t)\phi(s_t)$.

    (Hint: $w = \mathbb{E}[\phi\phi^T]^{-1}\mathbb{E}[\delta(\theta)\phi]$ because the convergence point w^* satisfies $\mathbb{E}[\delta\phi] = \mathbb{E}[\phi\phi^T]w^*$.)

- TDC algorithm: (C for correction)
  - **Objective**: $J(\theta) = \|V_\theta - \Pi_\mu T^\pi V_\theta\|^2_{d_\mu}$;
  - The deviation of objective is

$$-\frac{1}{2}\nabla J(\theta) = \mathbb{E}_{(s,a,r,s')\sim MC2_{\pi,\mu}}[\delta(s,a,r,s')\phi(s)] \\ - \gamma\mathbb{E}_{(s,a,r,s')\sim MC2_{\pi,\mu}}[\phi(s')\phi^T(s)] \\ \cdot \mathbb{E}_{s\sim d_\mu}[\phi(s)\phi^T(s)]^{-1} \\ \cdot \mathbb{E}_{(s,a,r,s')\sim MC2_{\pi,\mu}}[\delta(s,a,r,s')\phi(s)].$$

  - Algorithm step:
    - $\theta_{t+1} = \theta_t + \alpha_t\rho_t(\delta(s_t, a_t, r_t, s_{t+1})\phi(s_t) - \gamma\phi(s_{t+1})\phi^T(s_t)w_t)$;
    - $w_{t+1} = w_t + \beta_t(\rho_t\delta(s_t, a_t, r_t, s_{t+1}) - \phi^T(s_t)w_t)\phi_t$.
- Proximal GTD algorithm.

3. Policy Gradient(Policy Improvement, Actor)

3.1 The Objective of Policy Gradient

- $J(\theta) = \mathbb{E}[R(\tau)|\tau \sim MC_\pi]$

  then $\nabla_\theta\mathbb{E}_{\tau\sim MDP(\pi_\theta)}[R(\tau)] = \mathbb{E}_\tau\left[\sum_{t=0}^\infty \gamma^t\nabla_\theta\log\pi_\theta(a_t|s_t)\left(\sum_{t'=t}^\infty \gamma^{t'}r_{t'} - b(s_t)\right)\right]$;
- $J(\theta) = \mathbb{E}[V^{\pi_\theta}(s)|s\sim p] = \sum_{s\in S} p(s)i(s)V^{\pi_\theta}(s)$, where $V^{\pi_\theta}(s) = \mathbb{E}[\sum_{t=0}^T \gamma^t r_t] = \mathbb{E}[R(\tau)|s_0 = s]$;

  then $\nabla_\theta J(\theta)^T = p_i^T(I - P_{\pi,\gamma})^{-1}G$, where $G(s) = \left(\sum_a \frac{\partial\pi(s,a;\theta)}{\partial\theta}Q_\pi(s,a)\right)^T$;
  - $p = p_0$ and $\forall s, i(s) = 1$, then $\nabla_\theta J(\theta)$ is on-policy policy gradient in trajectory space;
  - $p = d_\pi$ and $\forall s, i(s) = 1$, then

    $\nabla_\theta J(\theta) = \frac{1}{1-\gamma}G^T d_\pi = \frac{1}{1-\gamma}\sum_s d_\pi(s)\sum_a \frac{\partial\pi(s,a;\theta)}{\partial\theta}Q_\pi(s,a)$,

    which is on-policy gradient in state-transition space;

    (Hint: If $I - A$ is invertible, then $(I-A)^{-1} = I + A + A^2 + \cdots$).
- The difficulty of off-policy policy gradient:

  $p = d_\mu$ and $\forall s, i(s) = 1$, then $\nabla_\theta J(\theta)^T = d_\mu^T(I - P_{\pi,\gamma})^{-1}G$;

3.2 Emphatic weightings method

- Emphatic weightings method: $M = d_\mu^T(I - P_{\pi,\gamma})^{-1}$;

  > **Theorem 1** (Off-policy Policy Gradient Theorem).

> $$\frac{\partial J_\mu(\theta)}{\partial\theta} = \sum_s m(s) \sum_a \frac{\partial \pi(s,a;\theta)}{\partial\theta} Q_\pi(s,a),$$
> where $m^T = i^T(I - P_{\pi,\gamma})^{-1}$, $i(s) = d_\mu(s)i(s)$ and
> $$P_{\pi,\gamma}(s,s') = \sum_a \pi(s,a;\theta)P(s,a,s')\gamma(s,a,s').$$

- The Algorithm steps:
  - $M_t = \gamma\rho_{t-1}M_{t-1} + 1$
  - $\theta \leftarrow \theta + \alpha_t\rho_t M_t \frac{\partial \ln \pi(s,a;\theta)}{\partial\theta} q_\pi(s,a)$;

3.3 Covariate Shift Method(COP)

COP-TD learning rule

- (Covariate Shift) Estimate $\tilde{M} \approx diag\left(\frac{d_\pi(s_1)}{d_\mu(s_1)}, \frac{d_\pi(s_2)}{d_\mu(s_2)}, \ldots, \frac{d_\pi(s_n)}{d_\mu(s_n)}\right)$, and let $d_\mu^T\tilde{M} \approx d_\pi^T$.
- We use $c(s) \approx \frac{d_\pi(s)}{d_\mu(s)}$ by td algorithm:$c(s') = c(s') + \alpha\left[\frac{\pi(a|s)}{\mu(a|s)}c(s) - c(s')\right]$, which corresponding the transition:

$$(Yc)(s') := \mathbb{E}_{s\sim d_\mu, a\sim\mu}\left[\frac{\pi(a|s)}{\mu(a|s)}c(s)\Big|s'\right].$$

The discounted COP-TD learning rule

$$c(s') = c(s') + \alpha\left[\hat{\gamma}\frac{\pi(a|s)}{\mu(a|s)}c(s) + (1-\hat{\gamma}) - c(s')\right].$$

The corresponding operator is

$$Y_{\hat{\gamma}}c := \hat{\gamma}Yc + (1-\hat{\gamma})e.$$

Definition 2. For a given $\hat{\gamma} \in [0,1]$, we define the discounted rest transition function $\hat{P}_\pi$ as:

$$\hat{P}_\pi := \hat{\gamma}P_\pi + (1-\hat{\gamma})ed_\mu^T.$$

The corresponding stationary distribution is $\hat{d}_\pi = \hat{d}_\pi = (1-\hat{\gamma})(I - \hat{\gamma}P_\pi^T)^{-1}d_\mu$.

$$c(s) \approx \frac{\hat{d}_\pi(s)}{d_\mu(s)}.$$

3.4 The General Policy Gradient

$$J_{\hat{\gamma}}(\theta) = \mathbb{E}[V^{\pi_\theta}(s)|s \sim \hat{d}_\pi] = \sum_{s\in S}\hat{d}_\pi\hat{i}(s)V^{\pi_\theta}(s).$$

which corresponding the space $MC2_{\pi,\hat{\pi}}$.

Theorem 1(Generalized Off-Policy Policy Gradient Theorem)

$$\nabla J_{\hat{\gamma}}(\theta) = \sum_s m(s)\sum_a q_\pi(s,a)\nabla_\theta\pi(a|s) + \sum_s d_\mu(s)\hat{i}(s)v_\pi(s)g(s),$$

where $g = \hat{\gamma}D_\mu^{-1}(I - \hat{\gamma}P_\pi^T)^{-1}b$ and $b = \nabla_\theta P_\pi^T D_\mu c$.

[1] Tsitsiklis J N, Van Roy B. An analysis of temporal-difference learning with function approximation[J]. IEEE Transactions on Automatic Control, 1997, 42(5): 674-690.

[2] Sutton R S, Mahmood A R, White M, et al. An emphatic approach to the problem of off-policy temporal-difference learning[J]. Journal of Machine Learning Research, 2016, 17(1): 2603-2631.

[3] Maei, H. R. (2011). Gradient Temporal-Difference Learning Algorithms. PhD thesis, University of Alberta.

[4] Imani E, Graves E, White M, et al. An Off-policy Policy Gradient Theorem Using Emphatic Weightings[C]. neural information processing systems, 2018: 96-106.

[5] Gelada C, Bellemare M G. Off-Policy Deep Reinforcement Learning by Bootstrapping the Covariate Shift[C]. national conference on artificial intelligence, 2019: 3647-3655.

[6] Zhang S, Boehmer W, Whiteson S, et al. Generalized Off-Policy Actor-Critic.[J]. arXiv: Learning, 2019.