Breaking the Deadly Triad with a Target Network

Shangtong Zhang [1] **Hengshuai Yao** [2,3] **Shimon Whiteson** [1]

Abstract

The deadly triad refers to the instability of a reinforcement learning algorithm when it employs off-policy learning, function approximation, and bootstrapping simultaneously. In this paper, we investigate the target network as a tool for breaking the deadly triad, providing theoretical support for the conventional wisdom that a target network stabilizes training. We first propose and analyze a novel target network update rule which augments the commonly used Polyak-averaging style update with two projections. We then apply the target network and ridge regularization in several divergent algorithms and show their convergence to regularized TD fixed points. Those algorithms are off-policy with linear function approximation and bootstrapping, spanning both policy evaluation and control, as well as both discounted and average-reward settings. In particular, we provide the first convergent linear Q-learning algorithms under nonrestrictive and changing behavior policies without bi-level optimization.

1. Introduction

The deadly triad (see, e.g., Chapter 11.3 of Sutton & Barto (2018)) refers to the instability of a value-based reinforcement learning (RL, Sutton & Barto (2018)) algorithm when it employs off-policy learning, function approximation, and bootstrapping simultaneously. Different from *on-policy* methods, where the policy of interest is executed for data collection, *off-policy* methods execute a different policy for data collection, which is usually safer (Dulac-Arnold et al., 2019) and more data efficient (Lin, 1992; Sutton et al., 2011). *Function approximation* methods use parameterized functions, instead of a look-up table, to represent quantities of interest, which usually cope better with large-scale problems (Mnih et al., 2015; Silver et al., 2016). *Bootstrapping* methods construct update targets for an estimate by using the estimate itself recursively, which usually has lower variance than *Monte Carlo* methods (Sutton, 1988). However, when an algorithm employs all those three preferred ingredients (off-policy learning, function approximation, and bootstrapping) simultaneously, there is usually no guarantee that the resulting algorithm is well behaved and the value estimates can easily diverge (see, e.g., Baird (1995); Tsitsiklis & Van Roy (1997); Zhang et al. (2021)), yielding the notorious deadly triad.

An example of the deadly triad is Q-learning (Watkins & Dayan, 1992) with linear function approximation, whose divergence is well documented in Baird (1995). However, Deep-Q-Networks (DQN, Mnih et al. (2015)), a combination of Q-learning and deep neural network function approximation, has enjoyed great empirical success. One major improvement of DQN over linear Q-learning is the use of a target network, a copy of the neural network function approximator (the main network) that is periodically synchronized with the main network. Importantly, the bootstrapping target in DQN is computed via the target network instead of the main network. As the target network changes slowly, it provides a stable bootstrapping target which in turn stabilizes the training of DQN. Instead of the periodical synchronization, Lillicrap et al. (2015) propose a Polyak-averaging style target network update, which has also enjoyed great empirical success (Fujimoto et al., 2018; Haarnoja et al., 2018).

Inspired by the empirical success of the target network in RL with deep networks, in this paper, we theoretically investigate the target network as a tool for breaking the deadly triad. We consider a two-timescale framework, where the main network is updated faster than the target network. By using a target network to construct the bootstrapping target, the main network update becomes least squares regression. After adding ridge regularization (Tikhonov et al., 2013) to this least squares problem, we show convergence for both the target and main networks.

Our main contributions are twofold. First, we propose a novel target network update rule augmenting the Polyak-averaging style update with two projections. The balls for the projections are usually large so most times they are just identity mapping. However, those two projections offer sig-

[1]University of Oxford [2]Huawei Technologies [3]University of Alberta. Correspondence to: Shangtong Zhang <shangtong.zhang@cs.ox.ac.uk>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

nificant theoretical advantages making it possible to analyze where the target network converges to (Section 3). Second, we apply the target network in various existing divergent algorithms and show their convergence to regularized TD (Sutton, 1988) fixed points. Those algorithms are off-policy algorithms with linear function approximation and bootstrapping, spanning both policy evaluation and control, as well as both discounted and average-reward settings. In particular, we provide the first convergent linear Q-learning algorithms under nonrestrictive and changing behavior policies without bi-level optimization, for both discounted and average-reward settings.

2. Background

Let M be a real positive definite matrix and x be a vector, we use $\|x\|_M \doteq \sqrt{x^\top M x}$ to denote the norm induced by M and $\|\cdot\|_M$ to denote the corresponding induced matrix norm. When M is the identity matrix I, we ignore the subscript I for simplicity. We use vectors and functions interchangeably when it does not cause confusion, e.g., given $f : \mathcal{X} \to \mathbb{R}$, we also use f to denote the corresponding vector in $\mathbb{R}^{|\mathcal{X}|}$. All vectors are column vectors. We use $\mathbf{1}$ to denote an all one vector, whose dimension can be deduced from the context. $\mathbf{0}$ is similarly defined.

We consider an infinite horizon Markov Decision Process (MDP, see, e.g., Puterman (2014)) consisting of a finite state space $\mathcal{S}$, a finite action space $\mathcal{A}$, a transition kernel $p : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$, and a reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. At time step t, an agent at a state S_t executes an action $A_t \sim \pi(\cdot|S_t)$, where $\pi : \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the policy followed by the agent. The agent then receives a reward $R_{t+1} \doteq r(S_t, A_t)$ and proceeds to a new state $S_{t+1} \sim p(\cdot|S_t, A_t)$.

In the *discounted* setting, we consider a discount factor $\gamma \in [0, 1)$ and define the return at time step t as $G_t \doteq \sum_{i=1}^{\infty} \gamma^{i-1} R_{t+i}$, which allows us to define the action-value function $q_\pi(s, a) \doteq \mathbb{E}_{\pi,p}[G_t|S_t = s, A_t = a]$. The action-value function q_π is the unique fixed point of the Bellman operator $\mathcal{T}_\pi$, i.e., $q_\pi = \mathcal{T}_\pi q_\pi \doteq r + \gamma P_\pi q_\pi$, where $P_\pi \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}||\mathcal{A}|}$ is the transition matrix, i.e., $P_\pi((s, a), (s', a')) \doteq \sum_a p(s'|s, a)\pi(a'|s')$.

In the *average-reward* setting, we assume:

Assumption 2.1. *The chain induced by π is ergodic.*

This allows us to define the *reward rate* $\bar{r}_\pi \doteq \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[R_t|p, \pi]$. The differential action-value function $\bar{q}_\pi(s, a)$ is defined as

$$\lim_{T\to\infty} \sum_{t=0}^{T} \mathbb{E}_{\pi,p}[r(S_t, A_t) - \bar{r}_\pi|S_0 = s, A_0 = a].$$

The differential Bellman equation is

$$\bar{q} = r - \bar{r}\mathbf{1} + P_\pi \bar{q}, \tag{1}$$

where $\bar{q} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ and $\bar{r} \in \mathbb{R}$ are free variables. It is well known that all solutions to (1) form a set $\{(\bar{q}, \bar{r}) \mid \bar{r} = \bar{r}_\pi, \bar{q} = q_\pi + c\mathbf{1}, c \in \mathbb{R}\}$ (Puterman, 2014).

The *policy evaluation* problem refers to estimating q_π or $(\bar{q}_\pi, \bar{r}_\pi)$. The *control* problem refers to finding a policy π maximizing $q_\pi(s, a)$ for each (s, a) or maximizing $\bar{r}_\pi$. With *linear* function approximation, we approximate $q_\pi(s, a)$ or $\bar{q}_\pi(s, a)$ with $x(s, a)^\top w$, where $x : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^K$ is a feature mapping and $w \in \mathbb{R}^K$ is the learnable parameter. We use $X \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}| \times K}$ to denote the feature matrix, each row of which is $x(s, a)^\top$, and assume:

Assumption 2.2. *X has linearly independent columns.*

In the average-reward setting, we use an additional parameter $\bar{r} \in \mathbb{R}$ to approximate $\bar{r}_\pi$. In the *off-policy* learning setting, the data for policy evaluation or control is collected by executing a policy μ (behavior policy) in the MDP, which is different from π (target policy). In the rest of the paper, we consider the off-policy linear function approximation setting thus always assume $A_t \sim \mu(\cdot|S_t)$. We use as shorthand $x_t \doteq x(S_t, A_t), \bar{x}_t \doteq \sum_a \pi(a|S_t)x(S_t, a)$.

Policy Evaluation. In the discounted setting, similar to Temporal Difference Learning (TD, Sutton (1988)), one can use Off-Policy Expected SARSA to estimate q_π, which updates w as

$$\begin{aligned} \delta_t &\leftarrow R_{t+1} + \gamma \bar{x}_{t+1}^\top w_t - x_t^\top w_t, \\ w_{t+1} &\leftarrow w_t + \alpha_t \delta_t x_t, \end{aligned} \tag{2}$$

where $\{\alpha_t\}$ are learning rates. In the average-reward setting, (1) implies that $\bar{r}_\pi = d^\top(r + P_\pi \bar{q}_\pi - \bar{q}_\pi)$ holds for any probability distribution d. In particular, it holds for $d = d_\mu$. Consequently, to estimate $\bar{q}_\pi$ and $\bar{r}_\pi$, Wan et al. (2020); Zhang et al. (2021) update w and $\bar{r}$ as

$$\begin{aligned} w_{t+1} &\leftarrow w_t + \alpha_t(R_{t+1} - \bar{r}_t + \bar{x}_{t+1}^\top w_t - x_t^\top w_t)x_t, \\ \bar{r}_{t+1} &\leftarrow \bar{r}_t + \alpha_t(R_{t+1} + \bar{x}_{t+1}^\top w_t - x_t^\top w_t - \bar{r}_t). \end{aligned} \tag{3}$$

Unfortunately, both (2) and (3) can possibly diverge (see, e.g., Tsitsiklis & Van Roy (1997); Zhang et al. (2021)), which exemplifies the deadly triad in discounted and average-reward settings respectively.

Control. In the discounted setting, Q-learning with linear function approximation yields

$$\begin{aligned} \delta_t &\leftarrow R_{t+1} + \gamma \max_{a'} x(S_{t+1}, a')^\top w_t - x_t^\top w_t, \\ w_{t+1} &\leftarrow w_t + \alpha_t \delta_t x_t. \end{aligned} \tag{4}$$

In the average-reward setting, Differential Q-learning (Wan et al., 2020) with linear function approximation yields

$$\begin{aligned} &\delta_t \leftarrow R_{t+1} - \bar{r}_t + \gamma \max_{a'} x(S_{t+1}, a')^\top w_t - x_t^\top w_t, \\ &w_{t+1} \leftarrow w_t + \alpha_t \delta_t x_t, \quad \bar{r}_{t+1} \leftarrow \bar{r}_t + \alpha_t \delta_t. \end{aligned} \tag{5}$$

Unfortunately, both (4) and (5) can possibly diverge as well (see, e.g., Baird (1995); Zhang et al. (2021)), exemplifying the deadly triad again.

Motivated by the empirical success of the target network in deep RL, one can apply the target network in the linear function approximation setting. For example, using a target network in (4) yields

$$\delta_t \leftarrow R_{t+1} + \gamma \max_{a'} x(S_{t+1}, a')^\top \theta_t - x_t^\top w_t,$$

$$w_{t+1} \leftarrow w_t + \alpha_t \delta_t x_t, \tag{6}$$

$$\theta_{t+1} \leftarrow \theta_t + \beta_t (w_t - \theta_t), \tag{7}$$

where θ denotes the target network, $\{\beta_t\}$ are learning rates, and we consider the Polyak-averaging style target network update. The convergence of (6) and (7), however, remains unknown. Besides target networks, regularization has also been widely used in deep RL, e.g., Mnih et al. (2015) consider a Huber loss instead of a mean-squared loss; Lillicrap et al. (2015) consider ℓ_2 weight decay in updating Q-values.

3. Analysis of the Target Network

In Sections 4 & 5, we consider the merits of using a target network in several linear RL algorithms (e.g., (2) (3) (4) (5)). To this end, in this section, we start by proposing and analyzing a novel target network update rule:

$$\theta_{t+1} \doteq \Gamma_{B_1}\big(\theta_t + \beta_t(\Gamma_{B_2}(w_t) - \theta_t)\big). \tag{8}$$

In (8), w denotes the main network and θ denotes the target network. $\Gamma_{B_1} : \mathbb{R}^K \to \mathbb{R}^K$ is a projection to the ball $B_1 \doteq \big\{x \in \mathbb{R}^K \mid \|x\| \leq R_{B_1}\big\}$, i.e.,

$$\Gamma_{B_1}(x) \doteq x \mathbb{I}_{\|x\| \leq R_{B_1}} + (R_{B_1} x / \|x\|) \mathbb{I}_{\|x\| > R_{B_1}},$$

where $\mathbb{I}$ is the indicator function. Γ_{B_2} is a projection onto the ball B_2 with a radius R_{B_2}. We make the following assumption about the learning rates:

Assumption 3.1. *$\{\beta_t\}$ is a deterministic positive nonincreasing sequence satisfying $\sum_t \beta_t = \infty, \sum_t \beta_t^2 < \infty$.*

While (8) specifies only how θ is updated, we assume w is updated such that w can track θ in the sense that

Assumption 3.2. *There exists $w^* : \mathbb{R}^K \to \mathbb{R}^K$ such that $\lim_{t\to\infty} \|w_t - w^*(\theta_t)\| = 0$ almost surely.*

After making some additional assumptions on w^*, we arrive at our general convergent results.

Assumption 3.3. $\sup_\theta \|w^*(\theta)\| < R_{B_2} < R_{B_1} < \infty$.

Assumption 3.4. *w^* is a contraction mapping w.r.t. $\|\cdot\|$.*

Theorem 1. *(Convergence of Target Network) Under Assumptions 3.1-3.4, the iterate $\{\theta_t\}$ generated by (8) satisfies*

$$\lim_{t\to\infty} w_t = \lim_{t\to\infty} \theta_t = \theta^* \quad \textit{almost surely,}$$

where θ^* is the unique fixed point of $w^*(\cdot)$.

Assumptions 3.2 - 3.4 are assumed only for now. Once the concrete update rules for w are specified in the algorithms in Sections 4 & 5, we will prove that those assumptions indeed hold. Assumption 3.2 is expected to hold because we will later require that the target network to be updated much slower than the main network. Consequently, the update of the main network will become a standard least-square regression, whose solution w^* usually exists. Assumption 3.4 is expected to hold becuase we will later apply ridge regularization to the least-square regression. Consequently, its solution w^* will not change too fast w.r.t. the change of the regression target.

The target network update (8) is the same as that in (7) except for the two projections, where the first projection Γ_{B_1} is standard in optimization literature. The second projection Γ_{B_2}, however, appears novel and plays a crucial role in our analysis. *First*, if we have only Γ_{B_1}, the iterate $\{\theta_t\}$ would converge to the invariant set of the ODE

$$\tfrac{\mathrm{d}}{\mathrm{d}t}\theta(t) = w^*(\theta(t)) - \theta(t) + \zeta(t), \tag{9}$$

where $\zeta(t)$ is a reflection term that moves $\theta(t)$ back to B_1 when $\theta(t)$ becomes too large (see, e.g., Section 5 of Kushner & Yin (2003)). Due to this reflection term, it is possible that $\theta(t)$ visits the boundary of B_1 infinitely often. It thus becomes unclear what the invariant set of (9) is even if w^* is contractive. By introducing the second projection Γ_{B_2} and ensuring $R_{B_1} > R_{B_2}$, we are able to remove the reflection term and show that the iterate $\{\theta_t\}$ tracks the ODE

$$\tfrac{\mathrm{d}}{\mathrm{d}t}\theta(t) = w^*(\theta(t)) - \theta(t), \tag{10}$$

whose invariant set is a singleton $\{\theta^*\}$ when Assumption 3.4 holds. See the proof of Theorem 1 in Section A.1 based on the ODE approach (Kushner & Yin, 2003; Borkar, 2009) for more details. *Second*, to ensure the main network tracks the target network in the sense of Assumption 3.2 in our applications in Sections 4 & 5, it is crucial that the target network changes sufficiently slowly in the following sense:

Lemma 1. $\|\theta_{t+1} - \theta_t\| \leq \beta_t C_0$ *for some constant* $C_0 > 0$.

Lemma 1 would not be feasible without the second projection Γ_{B_2} and we defer its proof to Section A.2

In Sections 4 & 5, we provide several applications of Theorem 1 in both discounted and average-reward settings, for both policy evaluation and control. We consider a two-timescale framework, where the target network is updated more slowly than the main network. Let $\{\alpha_t\}$ be the learning rates for updating the main network w; we assume

Assumption 3.5. *$\{\alpha_t\}$ is a deterministic positive nonincreasing sequence satisfying $\sum_t \alpha_t = \infty, \sum_t \alpha_t^2 < \infty$. Further, for some $d > 0$, $\sum_t (\beta_t/\alpha_t)^d < \infty$.*

4. Application to Off-Policy Policy Evaluation

In this paper, we consider estimating the action-value q_π instead of the state-value v_π for unifying notations of policy evaluation and control. The algorithms for estimating v_π are straightforward up to change of notations and introduction of importance sampling ratios.

Discounted Setting. Using a target network for bootstrapping in (2) yields

$$w_{t+1} \leftarrow w_t + \alpha_t(R_{t+1} + \gamma \bar{x}_{t+1}^\top \theta_t - x_t^\top w_t)x_t.$$

As θ_t is quasi-static for w_t (Lemma 1 and Assumption 3.5), this update becomes least squares regression. Motivated by the success of ridge regularization in least squares and the widespread use of weight decay in deep RL, which is essentially ridge regularization, we add ridge regularization to this least squares, yielding Q-evaluation with a Target Network (Algorithm 1).

Algorithm 1 Q-evaluation with a Target Network

INPUT: $\eta > 0, R_{B_1} > R_{B_2} > 0$
Initialize $\theta_0 \in B_1$ and S_0
Sample $A_0 \sim \mu(\cdot|S_0)$
for $t = 0, 1, \dots$ **do**
Execute A_t, get R_{t+1} and S_{t+1}
Sample $A_{t+1} \sim \mu(\cdot|S_{t+1})$
$\bar{x}_{t+1} \doteq \sum_{a'} \pi(a'|S_{t+1})x(S_{t+1}, a')$
$\delta_t \doteq R_{t+1} + \gamma \bar{x}_{t+1}^\top \theta_t - x_t^\top w_t$
$w_{t+1} \doteq w_t + \alpha_t \delta_t x_t - \alpha_t \eta w_t$
$\theta_{t+1} \doteq \Gamma_{B_1}\big(\theta_t + \beta_t(\Gamma_{B_2}(w_t) - \theta_t)\big)$
end for

Let $A = X^\top D_\mu (I - \gamma P_\pi)X, b = X^\top D_\mu r$, where D_μ is a diagonal matrix whose diagonal entry is d_μ, the stationary state-action distribution of the chain induced by μ. Let $\Pi_{D_\mu} \doteq X(X^\top D_\mu X)^{-1} X^\top D_\mu$ be the projection to the column space of X. We have

Assumption 4.1. *The chain in $\mathcal{S} \times \mathcal{A}$ induced by μ is ergodic.*

Theorem 2. *Under Assumptions 2.2, 3.1, 3.5, & 4.1, for any $\xi \in (0, 1)$, let $C_0 \doteq \frac{2(1-\xi)\sqrt{\eta}}{\gamma \|P_\pi\|_{D_\mu}}, C_1 \doteq \frac{\|r\|}{2\xi\sqrt{\eta}} + 1$, then for all $\|X\| < C_0, C_1 < R_{B_1}, R_{B_1} - \xi < R_{B_2} < R_{B_1}$ the iterate $\{w_t\}$ generated by Algorithm 1 satisfies*

$$\lim\nolimits_{t\to\infty} w_t = w_\eta^* \quad \textit{almost surely,}$$

where w_η^* is the unique solution of $(A + \eta I)w - b = \mathbf{0}$, and

$$\begin{aligned}&\left\|Xw_\eta^* - q_\pi\right\| \\ \leq &\Big(\tfrac{\sigma_{\max}(X)^2}{\sigma_{\min}(X)^4 \sigma_{\min}(D_\mu)^{2.5}} \|q_\pi\|\eta + \left\|\Pi_{D_\mu} q_\pi - q_\pi\right\|\Big)/\xi,\end{aligned}$$

where $\sigma_{\max}(\cdot), \sigma_{\min}(\cdot)$ denotes the largest and minimum singular values.

We defer the proof to Section A.3. Theorem 2 requires that the balls for projection are sufficiently large, which is completely feasible in practice. Theorem 2 also requires that the feature norm $\|X\|$ is not too large. Similar assumptions on feature norms also appear in Zou et al. (2019); Du et al. (2019); Chen et al. (2019b); Carvalho et al. (2020); Wang & Zou (2020); Wu et al. (2020) and can be easily achieved by scaling.

The solutions to $Aw - b = \mathbf{0}$, if they exist, are TD fixed points for off-policy policy evaluation in the discounted setting (Sutton et al., 2009b;a). Theorem 2 shows that Algorithm 1 finds a regularized TD fixed point w_η^*, which is also the solution of Least-Squares TD methods (LSTD, Boyan (1999); Yu (2010)). LSTD maintains estimates for A and b (referred to as $\hat{A}$ and $\hat{b}$) in an online fashion, which requires $\mathcal{O}(K^2)$ computational and memory complexity per step. As $\hat{A}$ is not guaranteed to be invertible, LSTD usually uses $(\hat{A} + \eta I)^{-1}\hat{b}$ as the solution and η plays a key role in its performance (see, e.g, Chapter 9.8 of Sutton & Barto (2018)). By contrast, Algorithm 1 finds the LSTD solution (i.e., w_η^*) with only $\mathcal{O}(K)$ computational and memory complexity per step. Moreover, Theorem 2 provides a performance bound for w_η^*. Let $w_0^* \doteq A^{-1}b$; Kolter (2011) shows with a counterexample that the approximation error of TD fixed points (i.e., $\|Xw_0^* - q_\pi\|$) can be arbitrarily large if μ is far from π, as long as there is representation error (i.e., $\left\|\Pi_{D_\mu} q_\pi - q_\pi\right\| > 0$) (see Section 6 for details). By contrast, Theorem 2 guarantees that $\left\|Xw_\eta^* - q_\pi\right\|$ is bounded from above, which is one possible advantage of regularized TD fixed points.

Algorithm 2 Diff. Q-evaluation with a Target Network

INPUT: $\eta > 0, R_{B_1} > R_{B_2} > 0$
Initialize $[\theta_0^r, \theta_0^{w\top}]^\top \in B_1$ and S_0
Sample $A_0 \sim \mu(\cdot|S_0)$
for $t = 0, 1, \dots$ **do**
Execute A_t, get R_{t+1} and S_{t+1}
Sample $A_{t+1} \sim \mu(\cdot|S_{t+1})$
$\bar{x}_{t+1} \doteq \sum_{a'} \pi(a'|S_{t+1})x(S_{t+1}, a')$
$\delta_t \doteq R_{t+1} - \theta_t^r + \bar{x}_{t+1}^\top \theta_t^w - x_t^\top w_t$
$w_{t+1} \doteq w_t + \alpha_t \delta_t x_t - \alpha_t \eta w_t$
$\bar{r}_{t+1} \doteq \bar{r}_t + \alpha_t(R_{t+1} + \bar{x}_{t+1}^\top \theta_t^w - x_t^\top \theta_t^w - \bar{r}_t)$
$\begin{bmatrix}\theta_{t+1}^r \\ \theta_{t+1}^w\end{bmatrix} \doteq \Gamma_{B_1}\Big(\begin{bmatrix}\theta_t^r \\ \theta_t^w\end{bmatrix} + \beta_t(\Gamma_{B_2}(\begin{bmatrix}\bar{r}_t \\ w_t\end{bmatrix}) - \begin{bmatrix}\theta_t^r \\ \theta_t^w\end{bmatrix})\Big)$
end for

Average-reward Setting. In the average-reward setting, we need to learn both $\bar{r}$ and w. Hence, we consider target networks θ^r and θ^w for $\bar{r}$ and w respectively. Plugging θ^r and θ^w into (3) for bootstrapping yields Differential Q-evaluation with a Target Network (Algorithm 2), where $\{B_i\}$ are now balls in R^{K+1}. In Algorithm 2, we impose ridge regularization only on w as $\bar{r}$ is a scalar and thus does not have any representation capacity limit.

Theorem 3. *Under Assumptions 2.1, 2.2, 3.1, 3.5, & 4.1, for any $\xi \in (0,1)$, there exist constants C_0 and C_1 such that for all $\|X\| < C_0, C_1 < R_{B_1}, R_{B_1} - \xi < R_{B_2} < R_{B_1}$, the iterates $\{\bar{r}_t\}$ and $\{w_t\}$ generated by Algorithm 2 satisfy*

$$\lim_{t\to\infty} \bar{r}_t = d_\mu^\top (r + P_\pi X w_\eta^* - X w_\eta^*),$$
$$\lim_{t\to\infty} w_t = w_\eta^* \quad \textit{almost surely,}$$

where w_η^* is the unique solution of $(\bar{A} + \eta I)w - \bar{b} = \boldsymbol{0}$ with

$$\bar{A} \doteq X(D_\mu - d_\mu d_\mu^\top)(I - P_\pi)X,$$
$$\bar{b} \doteq X^\top (D_\mu - d_\mu d_\mu^\top) r.$$

If features are zero-centered (i.e., $X^\top d_\mu = \boldsymbol{0}$), then

$$\left\| X w_\eta^* - \bar{q}_\pi^c \right\| \leq \left(\tfrac{\sigma_{\max}(X)^2}{\sigma_{\min}(X)^4 \sigma_{\min}(D_\mu)^{2.5}} \|\bar{q}_\pi^c\| \eta + \left\| \Pi_{D_\mu} \bar{q}_\pi^c - \bar{q}_\pi^c \right\| \right)/\xi,$$
$$|\bar{r}_\eta^* - \bar{r}_\pi| \leq \left\| d_\mu^\top (P_\pi - I) \right\| \inf_c \left\| (X w_\eta^* - \bar{q}_\pi^c) \right\|,$$

where $\bar{q}_\pi^c \doteq \bar{q}_\pi + c\boldsymbol{1}$.

We defer the proof to Section A.4. As the differential Bellman equation (1) has infinitely many solutions for $\bar{q}$, all of which differ only by some constant offsets, we focus on analyzing the quality of $X w_\eta^*$ w.r.t. $\bar{q}_\pi^c$ in Theorem 3. The zero-centered feature assumption is also used in Zhang et al. (2021), which can be easily fulfilled in practice by subtracting all features with the estimated mean. In the on-policy case (i.e., $\mu = \pi$), we have $d_\mu^\top (P_\pi - I) = \boldsymbol{0}$, indicating $\bar{r}_\eta^* = \bar{r}_\pi$, i.e., the regularization on the value estimate does not pose any bias on the reward rate estimate.

As shown by Zhang et al. (2021), if the update (3) converges, it converges to w_0^*, the TD fixed point for off-policy policy evaluation in the average-reward setting, which satisfies $\bar{A} w_0^* + \bar{b} = 0$. Theorem 3 shows that Algorithm 2 converges to a regularized TD fixed point. Though Zhang et al. (2021) give a bound on $\|X w_0^* - \bar{q}_\pi^c\|$, their bound holds only if μ is sufficiently close to π. By contrast, our bound on w_η^* in Theorem 3 holds for all μ.

5. Application to Off-Policy Control

Discounted Setting. Introducing a target network and ridge regularization in (4) yields Q-learning with a Target Network (Algorithm 3), where the behavior policy μ_θ depends on θ through the action-value estimate $X\theta$ and can be any policy satisfying the following two assumptions.

Assumption 5.1. *Let $\mathcal{P}$ be the closure of $\left\{ P_{\mu_\theta} \mid \theta \in \mathbb{R}^K \right\}$. For any $P \in \mathcal{P}$, the Markov chain evolving in $\mathcal{S} \times \mathcal{A}$ induced by P is ergodic.*

Assumption 5.2. *$\mu_\theta(a|s)$ is Lipschitz continuous in θ.*

Assumption 5.1 is standard. When the behavior policy μ is fixed (independent of θ), the induced chain is usually assumed to be ergodic when analyzing the behavior of Q-learning (see, e.g., Melo et al. (2008); Chen et al. (2019b); Cai et al. (2019)). In Algorithm 3, the behavior policy μ_θ changes every step, so it is natural to assume that any of those behavior policies induces an ergodic chain. A similar assumption is also used by Zou et al. (2019) in their analysis of on-policy linear SARSA. Moreover, Zou et al. (2019) assume not only the ergodicity but also the *uniform* ergodicity of their sampling policies. Similarly, in Assumption 5.1, we assume ergodicity for not only all the transition matrices, but also their limits (c.f. the closure $\mathcal{P}$). A similar assumption is also used by Marbach & Tsitsiklis (2001) in their analysis of on-policy actor-critic methods. Assumption 5.2 can be easily fulfilled, e.g., by using a softmax policy w.r.t. $x(s,\cdot)^\top \theta$.

Algorithm 3 Q-learning with a Target Network

INPUT: $\eta > 0, R_{B_1} > R_{B_2} > 0$
Initialize $\theta_0 \in B_1$ and S_0
Sample $A_0 \sim \mu_{\theta_0}(\cdot|S_0)$
for $t = 0, 1, \ldots$ **do**
  Execute A_t, get R_{t+1} and S_{t+1}
  Sample $A_{t+1} \sim \mu_{\theta_t}(\cdot|S_t)$
  $\delta_t \doteq R_{t+1} + \gamma \max_{a'} x(S_{t+1}, a')^\top \theta_t - x_t^\top w_t$
  $w_{t+1} \doteq w_t + \alpha_t \delta_t x_t - \alpha_t \eta w_t$
  $\theta_{t+1} \doteq \Gamma_{B_1}\big(\theta_t + \beta_t(\Gamma_{B_2}(w_t) - \theta_t)\big)$
end for

Theorem 4. *Under Assumptions 2.2, 3.1, 3.5, 5.1, & 5.2, for any $\xi \in (0,1), R_{B_1} > R_{B_2} > R_{B_1} - \xi > 0$, there exists a constant C_0 such that for all $\|X\| < C_0$, the iterate $\{w_t\}$ generated by Algorithm 3 satisfies*

$$\lim_{t\to\infty} w_t = w_\eta^* \quad \textit{almost surely,}$$

where w_η^* is the unique solution of

$$(A_{\pi_w, \mu_w} + \eta I) w - b_{\mu_w} = \boldsymbol{0} \tag{11}$$

inside B_1. Here

$$A_{\pi_w,\mu_w} \doteq X^\top D_{\mu_w}(I - \gamma P_{\pi_w})X,\; b_{\mu_w} \doteq X^\top D_{\mu_w} r,$$

and π_w denotes the greedy policy w.r.t. $x(s,\cdot)^\top w$.

We defer the proof to Section A.5. Analogously to the policy evaluation setting, if we call the solutions of $A_{\pi_w,\mu_w} w - b_{\mu_w} = \boldsymbol{0}$ TD fixed points for control in the discounted setting, then Theorem 4 asserts that Algorithm 3 finds a regularized TD fixed point.

Algorithm 3 and Theorem 4 are significant in two aspects. *First*, in Algorithm 3, the behavior policy is a function of the target network and thus changes every time step. By contrast, previous work on Q-learning with function approximation (e.g., Melo et al. (2008); Maei et al. (2010); Chen et al. (2019b); Cai et al. (2019); Chen et al. (2019a); Lee & He (2019); Xu & Gu (2020); Carvalho et al. (2020); Wang & Zou (2020)) usually assumes the behavior policy is fixed. Though Fan et al. (2020) also adopt a changing behavior policy, they consider bi-level optimization. At each time step, the nested optimization problem must be solved exactly, which is computationally expensive and sometimes unfeasible. To the best of our knowledge, we are the first to analyze Q-learning with function approximation under a changing behavior policy and without nested optimization problems. Compared with the fixed behavior policy setting or the bi-level optimization setting, our two-timescale setting with a changing behavior policy is more closely related to actual practice (e.g., Mnih et al. (2015); Lillicrap et al. (2015)).

Second, Theorem 4 does not enforce any similarity between μ_θ and π_w; they can be arbitrarily different. By contrast, previous work (e.g., Melo et al. (2008); Chen et al. (2019b); Cai et al. (2019); Xu & Gu (2020); Lee & He (2019)) usually requires the strong assumption that the fixed behavior policy μ is sufficiently close to the target policy π_w. As the target policy (i.e., the greedy policy) can change every time step due to the changing action-value estimates, this strong assumption rarely holds. While some work removes this strong assumption, it introduces other problems instead. In Greedy-GQ, Maei et al. (2010) avoid this strong assumption by computing sub-gradients of an MSPBE objective $\text{MSPBE}(w) \doteq \|A_{\pi_w,\mu} w - b_\mu\|^2_{C_\mu^{-1}}$ directly, where $C_\mu \doteq X^\top D_\mu X$. If linear Q-learning (4) under a fixed behavior policy μ converges, it converges to the minimizer of $\text{MSPBE}(w)$. Greedy-GQ, however, converges only to a stationary point of $\text{MSPBE}(w)$. By contrast, Algorithm 3 converges to a minimizer of our regularized MSPBE (c.f. (11)). In Coupled Q-learning, Carvalho et al. (2020) avoid this strong assumption by using a target network as well, which they update as

$$\theta_{t+1} \leftarrow \theta_t + \alpha_t((x_t x_t^\top) w_t - \theta_t). \tag{12}$$

This target network update deviates much from the commonly used Polyak-averaging style update, while our (8) is identical to the Polyak-averaging style update most times if the balls for projection are sufficiently large. Coupled Q-learning updates the main network w as usual (see (6)). With the Coupled Q-learning updates (6) and (12), Carvalho et al. (2020) prove that the main network and the target network converge to $\bar{w}$ and $\bar{\theta}$ respectively, which satisfy

$$X\bar{w} = XX^\top D_\mu \mathcal{T}_{\pi_{\bar{w}}} X\bar{w}, \quad X\bar{\theta} = \Pi_{D_\mu} \mathcal{T}_{\pi_{\bar{w}}} X\bar{w}.$$

It is, however, not clear how $\bar{w}$ and $\bar{\theta}$ relate to TD fixed points. Yang et al. (2019) also use a target network to avoid this strong assumption. Their target network update is the same as (8) except that they have only one projection Γ_{B_1}. Consequently, they face the problem of the reflection term $\zeta(t)$ (c.f. (9)). They also assume the main network $\{w_t\}$ is always bounded, a strong assumption that we do not require. Moreover, they consider a fixed sampling distribution for obtaining i.i.d. samples, while our data collection is done by executing the changing behavior policy μ_θ in the MDP.

One limit of Theorem 4 is that the bound on $\|X\|$ (i.e., C_0) depends on $1/R_{B_1}$ (see the proof in Section A.5 for the analytical expression), which means C_0 could potentially be small. Though we can use a small η accordingly to ensure that the regularization effect of η is modest, a small C_0 may not be desirable in some cases. To address this issue, we propose Gradient Q-learning with a Target Network, inspired by Greedy-GQ. We first equip $\text{MSPBE}(w)$ with a changing behavior policy μ_w, yielding the following objective $\|A_{\pi_w,\mu_w} w - b_{\mu_w}\|^2_{C_{\mu_w}^{-1}}$. We then use the target network θ in place of w in the non-convex components, yielding

$$L(w, \theta) \doteq \|A_{\pi_\theta,\mu_\theta} w - b_{\mu_\theta}\|^2_{C_{\mu_\theta}^{-1}} + \eta\|w\|^2, \tag{13}$$

where we have also introduced a ridge term. At time step t, we update w_t following the gradient $\nabla_w L(w, \theta_t)$ and update the target network θ_t as usual. Details are provided in Algorithm 4, where the additional weight vector $u \in \mathbb{R}^K$ results from a weight duplication trick (see Sutton et al. (2009b;a) for details) to address a double sampling issue in estimating $\nabla_w L(w, \theta)$.

Algorithm 4 Gradient Q-learning with a Target Network

INPUT: $\eta > 0, R_{B_1} > R_{B_2} > 0$
Initialize $\theta_0 \in B_1$ and S_0
Sample $A_0 \sim \mu_{\theta_0}(\cdot|S_0)$
for $t = 0, 1, \dots$ **do**
  Execute A_t, get R_{t+1} and S_{t+1}
  Sample $A_{t+1} \sim \mu_{\theta_t}(\cdot|S_t)$
  $\bar{x}_{t+1} \doteq \sum_{a'} \pi_{\theta_t}(a'|S_{t+1}) x(S_{t+1}, a')$
  $\delta_t \doteq R_{t+1} + \gamma \bar{x}_{t+1}^\top w_t - x_t^\top w_t$
  $u_{t+1} \doteq u_t + \alpha_t(\delta_t - x_t^\top u_t) x_t$
  $w_{t+1} \doteq w_t + \alpha_t(x_t - \gamma \bar{x}_{t+1}) x_t^\top u_t - \alpha_t \eta w_t$
  $\theta_{t+1} \doteq \Gamma_{B_1}\big(\theta_t + \beta_t(\Gamma_{B_2}(w_t) - \theta_t)\big)$
end for

In Algorithm 3, the target policy π_w is a greedy policy, which is not continuous in w. This discontinuity is not a problem there but requires sub-gradients in the analysis of Algorithm 4, which complicates the presentation. We, therefore, impose Assumption 5.2 on π_w as well.

Assumption 5.3. *$\pi_\theta(a|s)$ is Lipschitz continuous in θ.*

Though a greedy policy no longer satisfies Assumption 5.3, we can simply use a softmax policy.

Theorem 5. *Under Assumptions 2.2, 3.1, 3.5, & 5.1-5.3, there exist positive constants C_0 and C_1 such that for all $\|X\| < C_0, R_{B_1} > R_{B_2} > C_1$, the iterate $\{w_t\}$ generated by Algorithm 4 satisfies*

$$\lim_{t\to\infty} w_t = w_\eta^* \quad \textit{almost surely,}$$

where w_η^* is the unique solution of

$$(A_{\pi_w,\mu_w}^\top C_{\mu_w}^{-1} A_{\pi_w,\mu_w} + \eta I)w = A_{\pi_w,\mu_w}^\top C_{\mu_w}^{-1} b_{\mu_w}.$$

We defer the proof to Section A.6. Importantly, the C_0 here does not depend on R_{B_1} and R_{B_2}. More importantly, the condition on $\|X\|$ (or equivalently, η) in Theorem 5 is only used to fulfill Assumption 3.4, without which $\{\theta_t\}$ in Algorithm 4 still converges to an invariant set of the ODE (10). This condition is to investigate where the iterate converges to instead of whether it converges or not. If we assume $w_0^* \doteq \lim_{\eta\to 0} w_\eta^*$ exists and $A_{\pi_{w_0^*},\mu_{w_0^*}}$ is invertible, we can see $A_{\pi_{w_0^*},\mu_{w_0^*}} w_0^* - b_{\mu_{w_0^*}} = \mathbf{0}$, indicating w_0^* is a TD fixed point. w_η^* can therefore be regarded as a regularized TD fixed point, though how the regularization is imposed here (c.f. (13)) is different from that in Algorithm 3 (c.f. (11)).

Average-reward Setting. Similar to Algorithm 2, introducing a target network and ridge regularization in (5) yields Differential Q-learning with a Target Network (Algorithm 5). Similar to Algorithm 2, $\{B_i\}$ are now balls in R^{K+1}.

Algorithm 5 Diff. Q-learning with a Target Network

INPUT: $\eta > 0, R_{B_1} > R_{B_2} > 0$
Initialize $[\theta_0^r, \theta_0^{w\top}]^\top \in B_1$ and S_0
Sample $A_0 \sim \mu(\cdot|S_0)$
for $t = 0, 1, \ldots$ **do**
  Execute A_t, get R_{t+1} and S_{t+1}
  Sample $A_{t+1} \sim \mu_{\theta_t^w}(\cdot|S_{t+1})$
  $\delta_t \doteq R_{t+1} - \theta_t^r + \max_{a'} x(S_{t+1}, a')^\top \theta_t^w - x_t^\top w_t$
  $w_{t+1} \doteq w_t + \alpha_t \delta_t x_t - \alpha_t \eta w_t$
  $\delta_t' \doteq R_{t+1} + \max_{a'} x(S_{t+1}, a')^\top \theta_t^w - x_t^\top \theta_t^w - \bar{r}_t$
  $\bar{r}_{t+1} \doteq \bar{r}_t + \alpha_t \delta_t'$
  $\begin{bmatrix}\theta_{t+1}^r \\ \theta_{t+1}^w\end{bmatrix} \doteq \Gamma_{B_1}\left(\begin{bmatrix}\theta_t^r \\ \theta_t^w\end{bmatrix} + \beta_t\left(\Gamma_{B_2}\left(\begin{bmatrix}\bar{r}_t \\ w_t\end{bmatrix}\right) - \begin{bmatrix}\theta_t^r \\ \theta_t^w\end{bmatrix}\right)\right)$
end for

Theorem 6. *Under Assumptions 2.2, 3.1, 3.5, 5.1, & 5.2, let L_μ denote the Lipschitz constant of μ_θ, for any $\xi \in (0,1), R_{B_1} > R_{B_2} > R_{B_1} - \xi > 0$, there exist constants C_0 and C_1 such that for all $\|X\| < C_0, L_\mu < C_1$, the iterate $\{w_t\}$ generated by Algorithm 5 satisfies*

$$\lim_{t\to\infty} w_t = w_\eta^* \quad \textit{almost surely,}$$

where w_η^* is the unique solution of

$(\bar{A}_{\pi_w,\mu_w} + \eta I)w - \bar{b}_{\mu_w} = \mathbf{0}$ *inside B_1, where*

$$\bar{A}_{\pi_w,\mu_w} \doteq X(D_{\mu_w} - d_{\mu_w} d_{\mu_w}^\top)(I - P_{\pi_w})X,$$
$$\bar{b}_{\mu_w} \doteq X^\top(D_{\mu_w} - d_{\mu_w} d_{\mu_w}^\top)r,$$

and π_w is a greedy policy w.r.t. $x(s,\cdot)^\top w$.

We defer the proof to Section A.7. Theorem 6 requires μ_θ to be sufficiently smooth, which is a standard assumption even in the on-policy setting (e.g., Melo et al. (2008); Zou et al. (2019)). It is easy to see that if (5) converges, it converges to a solution of $\bar{A}_{\pi_w,\mu_w} w - \bar{b}_{\mu_w} = \mathbf{0}$, which we call a TD fixed point for control in the average-reward setting. Theorem 6, which shows that Algorithm 5 finds a regularized TD fixed point, is to the best of our knowledge the first theoretical study for linear Q-learning in the average-reward setting.

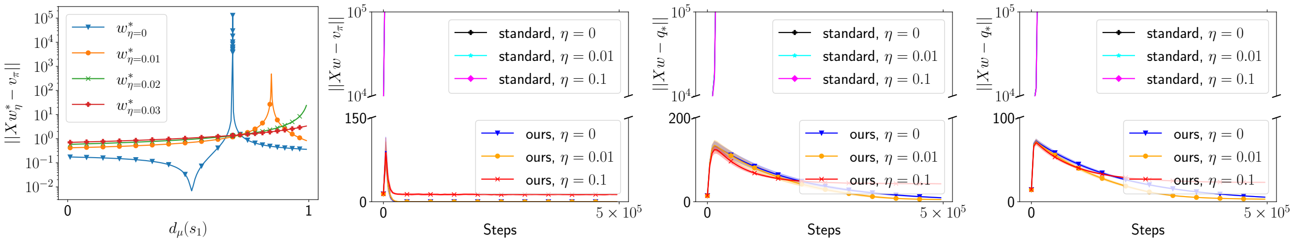
6. Experiments

All the implementations are publicly available. [1]

We first use Kolter's example (Kolter, 2011) to investigate how η influences the performance of w_η^* in the policy evaluation setting. Details are provided in Section D.1. This example is a two-state MDP with small representation error (i.e., $\left\|\Pi_{D_\mu} v_\pi - v_\pi\right\|$ is small). We vary the sampling probability of one state ($d_\mu(s_1)$) and compute corresponding w_η^* analytically. Figure 1a shows that with $\eta = 0$, the performance of w_η^* becomes arbitrarily poor when $d_\mu(s_1)$ approaches around 0.71. With $\eta = 0.01$, the spike exists as well. If we further increase η to 0.02 and 0.03, the performance for w_η^* becomes well bounded. This confirms the potential advantage of the regularized TD fixed points.

We then use Baird's example (Baird, 1995) to empirically investigate the convergence of the algorithms we propose. We use exactly the same setup as Chapter 11.2 of Sutton & Barto (2018). Details are provided in Section D.2. In particular, we consider three settings: policy evaluation (Figure 1b), control with a fixed behavior policy (Figure 1c), and control with an action-value dependent behavior policy (Figure 1d). For the policy evaluation setting, we compare a TD version of Algorithm 1 and standard Off-Policy Linear TD (possibly with ridge regularization). For the two control settings, we compare Algorithm 3 with standard linear Q-learning (possibly with ridge regularization). We use constant learning rates and do not use any projection in all the compared algorithms. The exact update rules are provided in Section D.2. Interestingly, Figures 1b-d show that even with $\eta = 0$, i.e., no ridge regularization, our algorithms with target network still converge in the tested domains. By contrast, without a target network, even when mild regularization is imposed, standard off-policy algorithms still diverge. This confirms the importance of the target network.

[1] https://github.com/ShangtongZhang/DeepRL



(a) (b) (c) (d)

Figure 1. (a) Effect of regularization on Kolter's example. v_π is the true state-value function. (b) Policy evaluation on Baird's example. (c) Control on Baird's example with a fixed behavior policy. (d) Control on Baird's example with an action-value-dependent behavior policy. In (b)(c)(d), the curves are averaged over 30 independent runs with shaded region indicating one standard deviation. q_* is the optimal action-value function. η is the weight for the ridge term. Those marked "ours" are curves of algorithms we propose; those marked "standard" are standard semi-gradient off-policy algorithms. Interestingly, the three "standard" curves overlap and get unbounded quickly.

7. Discussion and Related Work

For all the algorithms we propose, both the target network and the ridge regularization are at play. One may wonder if it is possible to ensure convergence with only ridge regularization without the target network. In the policy evaluation setting, the answer is affirmative. Applying ridge regularization in (2) directly yields

$$w_{t+1} \leftarrow w_t + \alpha_t \delta_t x_t - \alpha_t \eta w_t, \tag{14}$$

where δ_t is defined in (2). The expected update of (14) is

$$\begin{aligned}\Delta_w &\doteq b - (A + \eta I)w \\ &\doteq b - X^\top D_\mu X w + \gamma X^\top D_\mu (P_\pi X w) - \eta w.\end{aligned}$$

If its Jacobian w.r.t. w, denoted as $J_w(\Delta_w)$, is negative definite, the convergence of $\{w_t\}$ is expected (see, e.g., Section 5.5 of Vidyasagar (2002)). This negative definiteness can be easily achieved by ensuring $\eta > \|X\|^2 \|D_\mu(I - \gamma P_\pi)\|$ (see Diddigi et al. (2019) for similar techniques). This direct ridge regularization, however, would not work in the control setting. Consider, for example, linear Q-learning with ridge regularization (i.e., (14) with δ_t defined in (4)). The Jacobian of its expected update is $J_w(b_{\mu_w} - (A_{\pi_w,\mu_w} + \eta I)w)$. It is, however, not clear how to ensure this Jacobian is negative definite by tuning η. By using a target network for bootstrapping, $P_\pi X w$ becomes $P_\pi X \theta$. So $J_w(\Delta_w)$ becomes $-J_w(X^\top D_\mu X w + \eta w)$, which is always negative definite. Similarly, $J_w(b_{\mu_w} - (A_{\pi_w,\mu_w} + \eta)w)$ becomes $-J_w(X^\top D_{\mu_\theta} X w + \eta w)$ in Algorithm 3, which is always negative definite regardless of θ. The convergence of the main network $\{w_t\}$ can, therefore, be expected. The convergence of the target network $\{\theta_t\}$ is then delegated to Theorem 1. Now it is clear that in the deadly triad setting, the target network stabilizes training by ensuring the Jacobian of the expected update is negative definite. One may also wonder if it is possible to ensure convergence with only the target network without ridge regularization. The answer is unclear. In our analysis, the conditions on $\|X\|$ (or equivalently, η) are only sufficient and not necessarily necessary. We do see in Figure 1 that even with $\eta = 0$, our algorithms still converge in the tested domains. How small η can be in general and under what circumstances η can be 0 are still open problems, which we leave for future work. Further, ridge regularization usually affects the convergence rate of the algorithm, which we also leave for future work.

In this paper, we investigate target network as one possible solution for the deadly triad. Other solutions include Gradient TD methods (Sutton et al. (2009b;a; 2016) for the discounted setting; Zhang et al. (2021) for the average-reward setting) and Emphatic TD methods (Sutton et al. (2016) for the discounted setting). Other convergence results of Q-learning with function approximation include Tsitsiklis & Van Roy (1996); Szepesvári & Smart (2004), which require special approximation architectures, Wen & Van Roy (2013); Du et al. (2020), which consider deterministic MDPs, Li et al. (2011); Du et al. (2019), which require a special oracle to guide exploration, Chen et al. (2019a), which require matrix inversion every time step, and Wang et al. (2019); Yang & Wang (2019; 2020); Jin et al. (2020), which consider linear MDPs (i.e., both p and r are assumed to be linear). Achiam et al. (2019) characterize the divergence of Q-learning with nonlinear function approximation via Taylor expansions and use preconditioning to empirically stabilize training. Van Hasselt et al. (2018) empirically study the role of a target network in the deadly triad setting in deep RL, which is complementary to our theoretical analysis.

Regularization is also widely used in RL. Yu (2017) introduce a general regularization term to improve the robustness of Gradient TD algorithms. Du et al. (2017) use ridge regularization in MSPBE to improve its convexity. Zhang et al. (2020) use ridge regularization to stabilize the training of critic in an off-policy actor-critic algorithm. Kolter & Ng

(2009); Johns et al. (2010); Petrik et al. (2010); Painter-Wakefield et al. (2012); Liu et al. (2012) use Lasso regularization in policy evaluation, mainly for feature selection.

8. Conclusion

In this paper, we proposed and analyzed a novel target network update rule, with which we improved several linear RL algorithms that are known to diverge previously due to the deadly triad. Our analysis provided a theoretical understanding, in the deadly triad setting, of the conventional wisdom that a target network stabilizes training. A possibility for future work is to introduce nonlinear function approximation, possibly over-parameterized neural networks, into our analysis.

Acknowledgments

SZ is generously funded by the Engineering and Physical Sciences Research Council (EPSRC). SZ was also partly supported by DeepDrive. Inc from September to December 2020 during an internship. This project has received funding from the European Research Council under the European Union's Horizon 2020 research and innovation programme (grant agreement number 637713). The experiments were made possible by a generous equipment grant from NVIDIA.

References

Achiam, J., Knight, E., and Abbeel, P. Towards characterizing divergence in deep q-learning. *arXiv preprint arXiv:1903.08894*, 2019.

Baird, L. Residual algorithms: Reinforcement learning with function approximation. *Machine Learning*, 1995.

Borkar, V. S. *Stochastic approximation: a dynamical systems viewpoint*. Springer, 2009.

Boyan, J. A. Least-squares temporal difference learning. In *Proceedings of the 16th International Conference on Machine Learning*, 1999.

Cai, Q., Yang, Z., Lee, J. D., and Wang, Z. Neural temporal-difference and q-learning provably converge to global optima. *arXiv preprint arXiv:1905.10027*, 2019.

Carvalho, D., Melo, F. S., and Santos, P. A new convergent variant of q-learning with linear function approximation. *Advances in Neural Information Processing Systems*, 33, 2020.

Chen, S., Devraj, A. M., Bušić, A., and Meyn, S. Zap q-learning with nonlinear function approximation. *arXiv preprint arXiv:1910.05405*, 2019a.

Chen, Z., Zhang, S., Doan, T. T., Clarke, J.-P., and Maguluri, S. T. Finite-sample analysis of nonlinear stochastic approximation with applications in reinforcement learning. *arXiv preprint arXiv:1905.11425*, 2019b.

Diddigi, R. B., Kamanchi, C., and Bhatnagar, S. A convergent off-policy temporal difference algorithm. *arXiv preprint arXiv:1911.05697*, 2019.

Du, S. S., Chen, J., Li, L., Xiao, L., and Zhou, D. Stochastic variance reduction methods for policy evaluation. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

Du, S. S., Luo, Y., Wang, R., and Zhang, H. Provably efficient q-learning with function approximation via distribution shift error checking oracle. In *Advances in Neural Information Processing Systems*, 2019.

Du, S. S., Lee, J. D., Mahajan, G., and Wang, R. Agnostic q-learning with function approximation in deterministic systems: Near-optimal bounds on approximation error and sample complexity. *Advances in Neural Information Processing Systems*, 2020.

Dulac-Arnold, G., Mankowitz, D., and Hester, T. Challenges of real-world reinforcement learning. *arXiv preprint arXiv:1904.12901*, 2019.

Fan, J., Wang, Z., Xie, Y., and Yang, Z. A theoretical analysis of deep q-learning. In *Learning for Dynamics and Control*. PMLR, 2020.

Fujimoto, S., van Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. *arXiv preprint arXiv:1802.09477*, 2018.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pp. 2137–2143. PMLR, 2020.

Johns, J., Painter-Wakefield, C., and Parr, R. Linear complementarity for regularized policy evaluation and improvement. *Advances in neural information processing systems*, 2010.

Kolter, J. Z. The fixed points of off-policy td. In *Advances in Neural Information Processing Systems*, 2011.

Kolter, J. Z. and Ng, A. Y. Regularization and feature selection in least-squares temporal difference learning. In *Proceedings of the 26th annual international conference on machine learning*, 2009.

Konda, V. R. *Actor-critic algorithms*. PhD thesis, Massachusetts Institute of Technology, 2002.

Kushner, H. and Yin, G. G. *Stochastic approximation and recursive algorithms and applications*. Springer Science & Business Media, 2003.

Lee, D. and He, N. A unified switching system perspective and ode analysis of q-learning algorithms. *arXiv preprint arXiv:1912.02270*, 2019.

Li, L., Littman, M. L., Walsh, T. J., and Strehl, A. L. Knows what it knows: a framework for self-aware learning. *Machine learning*, 2011.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Lin, L.-J. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, 1992.

Liu, B., Mahadevan, S., and Liu, J. Regularized off-policy td-learning. *Advances in Neural Information Processing Systems*, 2012.

Maei, H. R., Szepesvári, C., Bhatnagar, S., and Sutton, R. S. Toward off-policy learning control with function approximation. In *Proceedings of the 27th International Conference on Machine Learning*, 2010.

Marbach, P. and Tsitsiklis, J. N. Simulation-based optimization of markov reward processes. *IEEE Transactions on Automatic Control*, 2001.

Melo, F. S., Meyn, S. P., and Ribeiro, M. I. An analysis of reinforcement learning with function approximation. In *Proceedings of the 25th International Conference on Machine Learning*, 2008.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *Nature*, 2015.

Painter-Wakefield, C., Parr, R., and Durham, N. L1 regularized linear temporal difference learning. *Technical report: Department of Computer Science, Duke University, Durham, NC, TR-2012–01*, 2012.

Petrik, M., Taylor, G., Parr, R., and Zilberstein, S. Feature selection using regularization in approximate linear programs for markov decision processes. *arXiv preprint arXiv:1005.1860*, 2010.

Puterman, M. L. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 2016.

Sutton, R. S. Learning to predict by the methods of temporal differences. *Machine Learning*, 1988.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction (2nd Edition)*. MIT press, 2018.

Sutton, R. S., Maei, H. R., Precup, D., Bhatnagar, S., Silver, D., Szepesvári, C., and Wiewiora, E. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th International Conference on Machine Learning*, 2009a.

Sutton, R. S., Maei, H. R., and Szepesvári, C. A convergent $o(n)$ temporal-difference algorithm for off-policy learning with linear function approximation. In *Advances in Neural Information Processing Systems*, 2009b.

Sutton, R. S., Modayil, J., Delp, M., Degris, T., Pilarski, P. M., White, A., and Precup, D. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems*, 2011.

Sutton, R. S., Mahmood, A. R., and White, M. An emphatic approach to the problem of off-policy temporal-difference learning. *The Journal of Machine Learning Research*, 2016.

Szepesvári, C. and Smart, W. D. Interpolation-based q-learning. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 100, 2004.

Tikhonov, A. N., Goncharsky, A., Stepanov, V., and Yagola, A. G. *Numerical methods for the solution of ill-posed problems*. Springer Science & Business Media, 2013.

Tsitsiklis, J. N. and Van Roy, B. Feature-based methods for large scale dynamic programming. *Machine Learning*, 1996.

Tsitsiklis, J. N. and Van Roy, B. Analysis of temporal-diffference learning with function approximation. In *Advances in Neural Information Processing Systems*, 1997.

Van Hasselt, H., Doron, Y., Strub, F., Hessel, M., Sonnerat, N., and Modayil, J. Deep reinforcement learning and the deadly triad. *arXiv preprint arXiv:1812.02648*, 2018.

Vidyasagar, M. *Nonlinear systems analysis*. SIAM, 2002.

Wan, Y., Naik, A., and Sutton, R. S. Learning and planning in average-reward markov decision processes. *arXiv preprint arXiv:2006.16318*, 2020.

Wang, Y. and Zou, S. Finite-sample analysis of greedy-gq with linear function approximation under markovian noise. *arXiv preprint arXiv:2005.10175*, 2020.

Wang, Y., Wang, R., Du, S. S., and Krishnamurthy, A. Optimism in reinforcement learning with generalized linear function approximation. *arXiv preprint arXiv:1912.04136*, 2019.

Watkins, C. J. and Dayan, P. Q-learning. *Machine Learning*, 1992.

Wen, Z. and Van Roy, B. Efficient exploration and value function generalization in deterministic systems. *Advances in Neural Information Processing Systems*, 2013.

Wu, Y., Zhang, W., Xu, P., and Gu, Q. A finite time analysis of two time-scale actor critic methods. *arXiv preprint arXiv:2005.01350*, 2020.

Xu, P. and Gu, Q. A finite-time analysis of q-learning with neural network function approximation. In *International Conference on Machine Learning*, pp. 10555–10565. PMLR, 2020.

Yang, L. and Wang, M. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In *International Conference on Machine Learning*, pp. 10746–10756. PMLR, 2020.

Yang, L. F. and Wang, M. Sample-optimal parametric q-learning using linearly additive features. *arXiv preprint arXiv:1902.04779*, 2019.

Yang, Z., Fu, Z., Zhang, K., and Wang, Z. Convergent reinforcement learning with function approximation: A bilevel optimization perspective, 2019. URL https://openreview.net/forum?id=ryfcCo0ctQ.

Yu, H. Convergence of least squares temporal difference methods under general conditions. In *ICML*, 2010.

Yu, H. On convergence of some gradient-based temporal-differences algorithms for off-policy learning. *arXiv preprint arXiv:1712.09652*, 2017.

Zhang, S., Liu, B., Yao, H., and Whiteson, S. Provably convergent two-timescale off-policy actor-critic with function approximation. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Zhang, S., Wan, Y., Sutton, R. S., and Whiteson, S. Average-reward off-policy policy evaluation with function approximation. *arXiv preprint arXiv:2101.02808*, 2021.

Zou, S., Xu, T., and Liang, Y. Finite-sample analysis for sarsa with linear function approximation. In *Advances in Neural Information Processing Systems*, 2019.