24th European Conference on Artificial Intelligence - ECAI 2020
Santiago de Compostela, Spain

A Convergent Off-Policy Temporal Difference Algorithm

Raghuram Bharadwaj Diddigi and **Chandramouli Kamanchi** and **Shalabh Bhatnagar** [1]

Abstract. Learning the value function of a given policy (target policy) from the data samples obtained from a different policy (behavior policy) is an important problem in Reinforcement Learning (RL). This problem is studied under the setting of off-policy prediction. Temporal Difference (TD) learning algorithms are a popular class of algorithms for solving the prediction problem. TD algorithms with linear function approximation are shown to be convergent when the samples are generated from the target policy (known as on-policy prediction). However, it has been well established in the literature that off-policy TD algorithms under linear function approximation may diverge. In this work, we propose a convergent on-line off-policy TD algorithm under linear function approximation. The main idea is to penalize the updates of the algorithm in a way as to ensure convergence of the iterates. We provide a convergence analysis of our algorithm. Through numerical evaluations, we further demonstrate the effectiveness of our algorithm.

1 Introduction

The two important problems in Reinforcement Learning (RL) [3] are Prediction and Control. The prediction problem deals with computing the value function of a given policy. In a discounted reward setting, value function refers to the total expected discounted reward obtained by following the given policy. The control problem refers to computing the optimal policy, i.e., the policy that maximizes the total expected discounted reward. When the model information (probability transition matrix and single-stage reward function) is fully known, techniques like value iteration and policy iteration are used to solve the control problem. Policy iteration is a two-step iterative algorithm where the task of prediction is performed in the first step for a given policy followed by the policy improvement task in the second step. However, in most of the practical scenarios, the model information is not known and instead, (state, action, reward and next-state) samples are only available. Under such a model-free setting, popular RL algorithms for prediction are Temporal Difference (TD) and for control are Q-Learning and Actor-Critic algorithms [17]. Actor-Critic algorithms can be seen as model-free analogs of the policy iteration algorithm and involve a model-free prediction step. Model-free prediction is an important problem for which optimal and convergent solutions are desired since it is also the stepping stone for the control problem.

TD algorithms under the tabular approach (where there is no approximation of the value function) are a popular class of algorithms for computing the exact value function of a given policy (henceforth referred to as target policy) from samples. In many of the real-life problems though, we encounter situations where the number of states is large or even infinite. In such cases, it is not possible to use tabular approaches and one has to resort to approximation based methods. TD algorithms are shown to be stable and convergent under linear function approximation, albeit under the setting of on-policy [3]. On-policy refers to the setting where state and action samples are obtained using the target policy itself. As we approach practical scenarios, it can be noted that such samples are not always available to the practitioner. For example, in games, say a practitioner would like to evaluate a (target) strategy. However, the data available might be from a player following a different strategy. The question that arises in this scenario is whether the practitioner can make use of this data and still evaluate the target strategy. These problems are studied under the setting of off-policy prediction where the goal is to evaluate the value function of the target policy from the data generated from a different policy (commonly referred to as behavior policy). The empirical success of the Deep Q-Learning algorithm [23] (a model-free control algorithm) motivates us to understand its convergence behavior, which is a very difficult problem. In fact, it has been noted in Section 11.3 of [17] that convergence and stability issues arise when we combine three components - function approximation, bootstrapping (TD algorithms) and off-policy learning, what they refer to as the “deadly triad”.

In our work, we propose an online off-policy stable TD algorithm for a prediction problem under linear function approximation. The idea is to penalize the parameters of the TD update to mitigate the divergence problem. We note here that the recent work [6] provides a comprehensive survey of algorithms for off-policy prediction problems and performs a comparative study. We now discuss some of the important and relevant works on the off-policy prediction problem.

In [4], Least-Squares TD algorithms (LSTD) with linear function approximation have been proposed that are shown to be convergent under both on-policy and off-policy settings. However, the per-step complexity of LSTD algorithms is quadratic in the number of parameters. In [15], off-policy TD algorithms are proposed that make use of an importance sampling idea to convert the expected value of total discounted reward under behavior policy to expected value under target policy. However, the variance of such algorithms is very high [19]. In [20], the Gradient TD (GTD) algorithm has been proposed that is stable under off-policy learning, linear function approximation and has linear (in the number of parameters) per-step complexity. Since then, there have been a lot of improvements on the GTD algorithm under various settings like prediction, control, and non-linear function approximation [12–14, 18]. The idea of adding the penalty in the form of a regularization term has been considered in [11] where Regularized off-policy TD (RO-TD) algorithm has been proposed based on GTD algorithms and convex-concave saddle point formulations. The regularization terms considered in the RL literature use the 2-norm or 1-norm or ∞-norm. However, in our

[1] Equal Contribution by the first two authors. All the authors are with the Department of Computer Science and Automation (CSA), Indian Institute of Science (IISc), Bangalore, India. Emails: {raghub,chandramouli,shalabh}@iisc.ac.in

setting, the regularization norm considered is the norm obtained from the stationary distribution of the Markov Chain realized by following the behavior policy. Emphatic TD algorithms (ETD) [7, 10, 19, 24] are another popular class of off-policy TD algorithms that achieve stability by emphasizing or de-emphasizing updates of the algorithm. These updates also have linear-time complexity. Moreover, these algorithms learn only one set of parameters, unlike GTD algorithms which are two-time scale stochastic approximation algorithms that learn two sets of parameters. Recently in [5, 9], a co-variance off-policy TD (COP-TD) algorithm has been proposed that includes a co-variance shift term in the TD update. This shift term is also learned along with the parameters of the algorithm.

Our algorithm, like the Emphatic TD algorithm, trains only one set of parameters and like ETD and GTD algorithms, has per-update complexity that is linear in the number of parameters. The contributions of our paper are as follows:

- We propose an on-line off-policy TD learning algorithm with linear function approximation. Our algorithm has linear per-iteration computational complexity in the number of parameters.
- We prove the convergence of our algorithm (Theorem 2 of Section 4) utilizing the techniques of [19, 21]. We also characterize the point of convergence of our algorithm (Section 5).
- We show the empirical performance of our algorithm and compare it with ETD and TDC (an algorithm from the GTD family) on standard benchmark off-policy divergent RL environments.

The rest of the paper is organized as follows. In Section 2, we introduce the background and preliminaries. We propose our algorithm in Section 3. Sections 4 and 5 describe the analysis of our algorithm. Section 6 presents the results of our numerical experiments. Finally, Section 7 presents concluding remarks and future research directions.

2 Background and Preliminaries

We consider a Markov Decision Process (MDP) given by (S, U, p, r, γ) where S denotes the state space. U is the set of actions, p is a probability transition rule where $p(s'|s, a)$ denotes the probability of transition to state s' when action a is chosen in state s. r is the single-stage reward function where $r(s, a)$ denotes the reward obtained by taking action a in state s. Finally, γ denotes the discount factor. Let $\pi : S \to \Delta(U)$ be the target policy where $\Delta(U)$ denotes the set of probability distributions over actions. The objective of the MDP prediction problem is to estimate the value function (V^π) of the target policy π, where the value function of a state $s \in S$ denoted by $V^\pi(s)$ is given by:

$$V^\pi(s) = \mathbb{E}\Big[\sum_{i=0}^{\infty} \gamma^i r(s_i, a_i)\Big| s_0 = s, \pi\Big], \qquad (1)$$

where the state-action trajectory $(s_0, a_0, s_1, \ldots)$ is obtained following the policy π and $\mathbb{E}[.]$ denotes the expectation over the trajectories.

As the number of states of the MDP can be very large, we resort to approximation techniques to compute the value function. In our work, we consider the linear function approximation architecture where

$$\widehat{V}(s) = \theta^T \phi(s), \qquad (2)$$

where $\widehat{V}(s)$ denotes the approximate value function associated with state s (that we desire to be very close to the exact value function), $\phi(s)$ is a $K \times 1$ feature vector associated with state s and θ is a $K \times 1$ weight vector. Note that the exact value function V^π may not be representable by (2). Therefore, the objective is to estimate the weight vector θ so that the approximate value function denoted by (2) is as close as possible to the exact value function.

The on-policy TD(0) [17] is a popular on-line algorithm for computing the weight vector θ. The update equation is given by:

$$\theta_{n+1} = \theta_n + \alpha_n (r_n + \gamma \theta_n^T \phi(s_{n+1}) - \theta_n^T \phi(s_n))\phi(s_n), \qquad (3)$$

where (s_n, r_n, s_{n+1}) is the state, reward and next state samples obtained at time n, $\alpha_n,\ n \geq 0$ is the step-size sequence and θ_0 denotes the initial parameter vector.

The stability of the on-policy TD(0) algorithm is well established in the literature [19]. We now outline the proof of convergence of this algorithm. Following the notation of [19], note that the update rule (3) can be re-written as:

$$\theta_{n+1} = \theta_n + \alpha_n (b_n - A_n \theta_n), \qquad (4)$$

where $A_n = \phi(s_n)(\phi(s_n) - \gamma\phi(s_{n+1}))^T$ and $b_n = r_{n+1}\phi(s_n)$.

It is shown in [21] that the algorithm with update rule (4) is stable if the matrix A given by:

$$A = \lim_{n \longrightarrow \infty} A_n = \Phi^T D_\pi (I - \gamma P_\pi)\Phi \qquad (5)$$

is positive definite. In (5), Φ is a $|S| \times K$ matrix with the feature vector $\phi(s)$ in row s. D_π is the $|S| \times |S|$ diagonal matrix with the diagonal being the stationary distribution of the Markov chain (assumed ergodic) obtained under policy π. Finally, P_π is a $|S| \times |S|$ matrix with $[P_\pi]_{ij} = \sum_a \pi(i, a)p(j|i, a)$. For the on-policy TD(0) algorithm, A is shown to be positive definite [19], thereby proving the stability of the algorithm.

In the off-policy prediction problem, the data samples are obtained from a behavior policy μ instead of the target policy π. In this case, the off-policy TD(0) update [19] is given by:

$$\theta_{n+1} = \theta_n + \alpha_n \rho_n \Big(r_n + \gamma \theta_n^T \phi(s_{n+1}) - \theta_n^T \phi(s_n)\Big)\phi(s_n), \qquad (6)$$

where r_n is the reward obtained by taking action a_n in state s_n and ρ_n is the importance sampling ratio given by $\frac{\pi(s_n, a_n)}{\mu(s_n, a_n)}$. The corresponding matrix A for this algorithm is given by:

$$A = \Phi^T D_\mu (I - \gamma P_\pi)\Phi, \qquad (7)$$

where D_μ is a diagonal matrix with diagonal being the stationary distribution of the Markov chain obtained under policy μ.

The matrix A defined in (7) need not be positive definite [19] in general. Therefore stability and convergence of the off-policy TD(0) are not guaranteed.

The off-policy TD(0) algorithm, if converges, may perform comparably to some of the off-policy convergent algorithms in the literature. For example, in Figure 5 of [6], it has been shown that the performance of off-policy TD(0) is comparable to that of the GTD(0) algorithm. However, as the algorithm is not stable, off-policy TD(0) can diverge. In this paper, we propose a simple and stable off-policy TD algorithm. In the next section, we propose our algorithm and in Section 4, we provide its convergence analysis.

3 The Proposed Algorithm

The input to our algorithm is the target policy, whose value function we want to estimate and the behavior policy, from which the samples are generated. Also, provided as an input to our algorithm is the

Algorithm 1 Perturbed Off-Policy Prediction Algorithm

```
    Input:
    $\mu, \pi$: behaviour and target policies respectively
    $(s_n, a_n, r_n, s_{n+1})_{n=0}^{\infty}$: data from behaviour policy
    $\theta_0$: initial parameter vector
    $\gamma$: discount factor
    $\phi(s)$: feature vector of state $s$
    $\eta$: non-negative perturbation parameter
    $\{\alpha_n\}$: step-size sequence
    Iter: total number of iterations
    Output: $\theta_{\text{Iter}}$
1:  procedure OFF-POLICY PREDICTION:
2:      while $n <$ Iter do
3:          $\rho_n = \frac{\pi(s_n,a_n)}{\mu(s_n,a_n)}$
4:          $\delta_n = r_n + \gamma\phi(s_{n+1})^T\theta_n - (1+\eta)\phi(s_n)^T\theta_n$
5:          $\theta_{n+1} = \theta_n + \alpha_n\rho_n\delta_n\phi(s_n)$
6:      return $\theta_{\text{Iter}}$
```

perturbation parameter ($\eta \geq 0$). The algorithm works as follows. At each time step n, we obtain a sample (s_n, a_n, r_n, s_{n+1}) using which importance sampling coefficient ρ_n is computed as shown in Step 3. We then compute our modified temporal difference term as shown in Step 4. Finally, the parameters of the algorithm are updated as shown in Step 5.

Remark 1 *From steps 3, 4 and 5 of the Algorithm 1, we infer that the per-step complexity is $\mathcal{O}(K)$, where K is the number of parameters.*

Remark 2 *The choice of η is critical in our algorithm. Larger values of η ensure convergence (see Theorem 2) and smaller values of η ensure more accurate solution (see Lemma 4).*

In the next section, we provide the convergence analysis of our proposed algorithm.

4 Convergence Analysis

In this section, the norms considered are as follows. For any vector $v \in \mathbb{R}^K$, $\|v\|_2$ denotes the 2-norm of v, that is $\|v\|_2 = \left(\sum_{i=1}^{K} v_i^2\right)^{\frac{1}{2}}$. Similarly for a $K \times K$ matrix A, $\|A\|$ denotes the norm induced by the 2-norm, that is $\|A\| = \sup\limits_{\|v\|_2=1} \dfrac{\|Av\|_2}{\|v\|_2}$. Also, given a diagonal matrix D with positive entries, we define $\|v\|_D$ as $\|v\|_D = \sqrt{v^T D v}$. Finally $\|v\|_\infty = \max\limits_{i=1}^{K} |v_i|$.

Now, the update rule of Algorithm 1 can be rewritten as follows.

$$\begin{aligned}\theta_{n+1} &= \theta_n + \alpha_n\rho_n\delta_n\phi(s_n)\\ &= \theta_n + \alpha_n(b_n - A_n\theta_n),\end{aligned}$$

where A_n and b_n are given by

$$A_n = -\rho_n\big(\gamma\phi(s_n)\phi(s_{n+1})^T - (1+\eta)\phi(s_n)\phi(s_n)^T\big), \tag{8}$$

$$b_n = \rho_n r_n \phi(s_n). \tag{9}$$

We state and invoke Theorem 2 of [21] (also see Th. 17, p. 239 of [2]) that we use to show the convergence of our algorithm.

Theorem 1 *Consider an iterative algorithm of the form*

$$\theta_{n+1} = \theta_n + \alpha_n\left(b(X_n) - A(X_n)\theta_n\right),$$

where

A1. *the step-size sequence satisfies $\sum\limits_{n=0}^{\infty}\alpha_n = \infty$, $\sum\limits_{n=0}^{\infty}\alpha_n^2 < \infty$.*

A2. *$X_n, n \geq 0$, is a Markov process with a unique stationary distribution.*

A3. *$A = \mathbb{E}_0[A(X_n)]$ and $b = \mathbb{E}_0[b(X_n)]$. Here $\mathbb{E}_0$ is the expectation with respect to the stationary distribution of the Markov chain $\{X_n\}$.*

A4. *The matrix A is positive definite.*

A5. *There exist positive constants C, q and a positive real valued function h from the states of the Markov chain $\{X_n\}$ such that $\sum\limits_{n=0}^{\infty}\|\mathbb{E}[A(X_n)|X_0 = X] - A\| \leq C\left(1 + h^q(X)\right)$ and*

$$\sum_{n=0}^{\infty}\|\mathbb{E}[b(X_n)|X_0 = X] - b\| \leq C\left(1 + h^q(X)\right).$$

A6. *For any $q > 1$ there exists a constant κ_q such that for all X and n, $\mathbb{E}[h^q(X_n)|X_0 = X] \leq \kappa_q\left(1 + h^q(X)\right)$.*

Under these assumptions, i.e., A1-A6 above, θ_n converges to the solution of $b - A\theta = 0$, almost surely.

To begin with, we define the process $\{X_n\}$ as follows. Let $X_n = (s_n, a_n, s_{n+1}), n \geq 0$. Observe that $\{X_n\}$ is a Markov chain as s_{n+1} is a deterministic function of X_n and the distribution of a_{n+1} and s_{n+2} depends only on s_{n+1}. Also note that, in our algorithm, $A(X_n) = A_n$ and $b(X_n) = b_n$ are given by equations (8) and (9) respectively with $X_n = (s_n, a_n, s_{n+1})$.

Several step-size sequences satisfy assumption A1, for e.g. $a_n = \frac{1}{n+1}, n \geq 0$. Assumption A2 is fairly general. For example, A2 holds under the assumption that the Markov chain $\{s_n\}$ from the policy μ is ergodic. We now validate assumptions A3, A4, A5 and A6 below. The assumption A3 is shown in Lemma 1. The assumption A4, i.e., the matrix A is positive definite, is shown in Theorem 2. Finally, assumptions A5 and A6 are shown to be true in Theorem 3.

We start by proving some important lemmas that are used in our main theorems.

Lemma 1 *Let Φ be the $|S| \times K$ matrix where the i^{th} row of Φ is given by $\phi(i)$, the feature vector of state i and r_π be the $|S| \times 1$ vector where the i^{th} component is given by $r_\pi(i) = \sum_{a\in U} r(i,a)\pi(i,a)$. Let $\mathbb{E}_0$ be the expectation with respect to the stationary distribution of the Markov chain $\{X_n\}$ and d_μ be the stationary distribution of the Markov chain $\{s_n\}$. Then $A = \mathbb{E}_0[A_n]$ and $b = \mathbb{E}_0[b_n]$ are given by*

$$\begin{aligned}A &= \Phi^T D_\mu\left((1+\eta)I - \gamma P_\pi\right)\Phi,\\ b &= \Phi^T D_\mu r_\pi,\end{aligned}$$

where D_μ is a diagonal matrix with the i^{th} diagonal element being $d_\mu(i)$, the steady state probability of $\{s_n\}$ being in state i under policy μ.

Proof:

$$\begin{aligned}&\mathbb{E}_0[A_n]\\ &= -\mathbb{E}_0\left[\rho_n\big(\gamma\phi(s_n)\phi(s_{n+1})^T - (1+\eta)\phi(s_n)\phi(s_n)^T\big)\right]\\ &= -\sum_{i,j\in S, a\in U}\mu(i,a)\left[\frac{\pi(i,a)}{\mu(i,a)}\big(\gamma\phi(i)\phi(j)^T d_\mu(i)p(j|i,a)\right.\\ &\qquad\qquad\left. - (1+\eta)d_\mu(i)\phi(i)\phi(i)^T\big)\right]\end{aligned}$$

$$
\begin{aligned}
&= -\sum_{i,j} \gamma d_\mu(i)\phi(i)\phi(j)^T p_\pi(j|i) \\
&\qquad\qquad\qquad + \sum_i d_\mu(i)(1+\eta)\phi(i)\phi(i)^T \\
&= \Phi^T D_\mu((1+\eta)I - \gamma P_\pi)\Phi.
\end{aligned}
$$

Similarly,

$$
\begin{aligned}
b &= \mathbb{E}_0[b_n] = \mathbb{E}_0[\rho_n r_n \phi(s_n)] \\
&= \sum_{i,j\in S, a\in U} d_\mu(i)\mu(i,a)p(j|i,a)\left[\frac{\pi(i,a)}{\mu(i,a)} r(i,a)\phi(i)\right] \\
&= \Phi^T D_\mu r_\pi,
\end{aligned}
$$

which proves the existence of A and b that in-turn validates assumption A3. □

Definition 1 *A $K \times K$ matrix M is positive definite if for all $0 \neq y \in \mathbb{R}^K$, $y^T M y > 0$.*

Lemma 2 *Given a $K \times K$ matrix M, M is positive definite iff the symmetric matrix $S = M + M^T$ is positive definite.*

Proof: For $0 \neq y \in \mathbb{R}^K$, observe that

$$y^T S y = y^T M y + y^T M^T y = 2y^T M y,$$

since $(y^T M y)^T = y^T M y$ as both are scalars and $y^T M^T y = (y^T M y)^T$. Hence S is positive definite if and only if M is positive definite. □

Theorem 2 *Suppose $M = D\left((1+\eta)I - \gamma P\right)$ where $D = diag(d)$ is a diagonal matrix with positive diagonal entries, $P = [p(j|i)]$ is a Markov matrix and $\eta > \max\left(\max_i \dfrac{\gamma d^T p(.|i)}{d_i} - 1, 0\right)$ and $0 < \gamma < 1$ are positive constants. Then $M = [m_{ij}]$ is positive definite.*

Proof: Consider the symmetric matrix $S = M + M^T$. From Lemma 2, it is enough to show that S is positive definite. Since S is symmetric, it is diagonalizable. Therefore it is enough to show that the eigen-values of S are positive. From the Gershgorin circle theorem (see [8]) for any eigen-value λ of S, there exists i such that

$$
\begin{aligned}
|\lambda - 2m_{ii}| &\leq \sum_{j\neq i} |m_{ij}| + \sum_{j\neq i} |m_{ji}| \\
\implies \lambda &\geq 2m_{ii} - \sum_{j\neq i} |m_{ij}| - \sum_{j\neq i} |m_{ji}|.
\end{aligned}
$$

Now, $m_{ii} = d_i\left((1+\eta) - \gamma p(i|i)\right)$ and for $i \neq j$ we have $m_{ij} = -d_i \gamma p(j|i)$. Therefore $m_{ii} - \sum_{j\neq i} |m_{ij}| = (1+\eta-\gamma)d_i$ and $m_{ii} - \sum_{j\neq i} |m_{ji}| = \left((1+\eta)d_i - \gamma d^T p(i|.)\right)$.

$$\implies \lambda \geq (1+\eta-\gamma)d_i + \Big((1+\eta)d_i - \gamma d^T p(i|.)\Big) > 0$$

from the hypothesis $\eta > \max_i \dfrac{\gamma d^T p(i|.)}{d_i} - 1$. Since λ is an arbitrary eigen-value, it is clear that every eigen-value of S is positive, i.e., S is positive definite. Hence M is positive definite. In particular, given the behaviour policy μ and the target policy π, there exists $\eta > 0$ such that $A = \Phi^T D_\mu \left((1+\eta)I - \gamma P_\pi\right)\Phi$ is positive definite, thereby satisfying assumption A4. □

The following lemma makes use of the arguments similar to those in Section VII of [21].

Lemma 3 *There exists positive constant C such that for any given initial state X,*

$$
\begin{aligned}
&\sum_{n=0}^{\infty} \|\mathbb{E}[A(X_n)|X_0 = X] - A\| \leq C \text{ and} \\
&\sum_{n=0}^{\infty} \|\mathbb{E}[b(X_n)|X_0 = X] - b\| \leq C.
\end{aligned}
$$

Proof: Let $d_\mu^n(s) = Pr(s_n = s|X_0 = X)$ for any $X = (s_0, a_0, s_1)$ and D_μ^n be the $|S| \times |S|$ diagonal matrix with the diagonal element $D_\mu^n(s,s) = d_\mu^n(s)$ for any given X. Since the Markov chain $\{X_n\}$ has an invariant distribution there exist [16] scalars $L > 0$ and $0 < \alpha < 1$ such that

$$|Pr(s_n = s|X_0 = X) - d_\mu(s)| \leq L\alpha^n, \ \forall\ X \text{ and } n \geq 0.$$

Therefore,

$$\|D_\mu^n - D_\mu\| \leq L\alpha^n.$$

Now,

$$
\begin{aligned}
&\mathbb{E}[A(X_n)|X_0 = X] \\
&= -\mathbb{E}\left[\rho_n\big(\gamma\phi(s_n)\phi(s_{n+1})^T - (1+\eta)\phi(s_n)\phi(s_n)^T\big)|X_0 = X\right] \\
&= -\sum_{i,j\in S, a\in U} d_\mu^n(i)\mu(i,a)p(j|i,a)\left[\frac{\pi(i,a)}{\mu(i,a)}\big(\gamma\phi(i)\phi(j)^T \right. \\
&\qquad\qquad\qquad\qquad\qquad\qquad \left. - (1+\eta)\phi(i)\phi(i)^T\big)\right] \\
&= -\sum_{i,j} \gamma d_\mu^n(i)\phi(i)\phi(j)^T p_\pi(j|i) + \sum_i d_\mu^n(i)(1+\eta)\phi(i)\phi(i)^T \\
&= \Phi^T D_\mu^n((1+\eta)I - \gamma P_\pi)\Phi.
\end{aligned}
$$

Therefore, with $G = \|(1+\eta)I - \gamma P_\pi\|$ we have

$$
\begin{aligned}
&\sum_{n=0}^{\infty} \|\mathbb{E}[A(X_n)|X_0 = X] - A\| \\
&= \sum_{n=0}^{\infty} \|\Phi^T(D_\mu^n - D_\mu)((1+\eta)I - \gamma P_\pi)\Phi\| \\
&\leq \sum_{n=0}^{\infty} K^2 \max_{k,j} |\phi_k^T(D_\mu^n - D_\mu)((1+\eta)I - \gamma P_\pi)\phi_j| \\
&\leq K^2 \max_k \|\phi_k\| G \max_j \|\phi_j\| \sum_{n=0}^{\infty} \|D_\mu^n - D_\mu\| \\
&\leq GK^2 \max_k \|\phi_k\|^2 \frac{L}{1-\alpha}.
\end{aligned}
$$

Similarly,

$$
\begin{aligned}
&\mathbb{E}[b(X_n)|X_0 = X] \\
&= \mathbb{E}[\rho_n r_n \phi(s_n)|X_0 = X] \\
&= \sum_{i,j\in S, a\in U} d_\mu^n(i)\mu(i,a)p(j|i,a)\left[\frac{\pi(i,a)}{\mu(i,a)} r(i,a)\phi(i)\right] \\
&= \Phi^T D_\mu^n r_\pi,
\end{aligned}
$$

where the i^{th} component of r_π is given by $r_\pi(i) = \sum_{a \in U} r(i,a)\pi(i,a)$. We then have

$$\begin{aligned}
&\mathbb{E}[b(X_n)|X_0 = X] - b \\
&= \mathbb{E}[\rho_n r_n \phi(s_n))|X_0 = X] - \Phi^T D_\mu r_\pi = \Phi^T (D_\mu^n - D_\mu) r_\pi
\end{aligned}$$

and

$$\begin{aligned}
&\sum_{n=0}^{\infty} \|\mathbb{E}[b(X_n)|X_0 = X] - b\| \\
&= \sum_{n=0}^{\infty} \|\Phi^T (D_\mu^n - D_\mu) r_\pi\| \\
&\leq \sum_{n=0}^{\infty} K \max_k |\phi_k^T (D_\mu^n - D_\mu) r_\pi| \\
&\leq K \max_k \|\phi_k\| \|r_\pi\| \sum_{n=0}^{\infty} \|D_\mu^n - D_\mu\| \\
&\leq K \max_k \|\phi_k\| \|r_\pi\| \frac{L}{1-\alpha}.
\end{aligned}$$

The choice

$$C = \max\left\{ GK^2 \max_k \|\phi_k\|^2 \frac{L}{1-\alpha}, K \max_k \|\phi_k\| \|r_\pi\| \frac{L}{1-\alpha} \right\}$$

proves the lemma. □

Theorem 3 *The assumptions A5 and A6 are valid.*

Proof: From Lemma 3 and with the choice of $\kappa_q = 1$, $h \equiv 1$ and $C = \max\left\{ GK^2 \max_k \|\phi_k\|^2 \frac{L}{1-\alpha}, K \max_k \|\phi_k\| \|r_\pi\| \frac{L}{1-\alpha} \right\}$ assumptions A5 and A6 are satisfied. □

Hence by Theorem 1, $\theta_n \rightarrow \theta^*$ almost surely, where $A\theta^* = b$. To describe the point of convergence of our algorithm consider for a given policy μ and a parameter η, $T_\mu^\eta : \mathbb{R}^{|S|} \rightarrow \mathbb{R}^{|S|}$ as $T_\mu^\eta = \frac{1}{1+\eta} T_\mu$. We state and prove the following properties about T_μ^η.

Lemma 4 *T_μ^η is a $\|.\|_\infty$-contraction and converges point-wise to T_μ as $\eta \rightarrow 0$.*

Proof: From the definition $T_\mu^\eta = \frac{1}{1+\eta} T_\mu$, for any $V \in \mathbb{R}^{|S|}$,

$$T_\mu^\eta V = \frac{1}{1+\eta} T_\mu V \rightarrow T_\mu V \text{ as } \eta \rightarrow 0.$$

Moreover, for any $V, W \in \mathbb{R}^{|S|}$,

$$\begin{aligned}
\|T_\mu^\eta V - T_\mu^\eta W\|_\infty =& \frac{\gamma}{1+\eta} \|P_\mu (V - W)\|_\infty \\
\leq& \frac{\gamma}{1+\eta} \|V - W\|_\infty.
\end{aligned}$$

Hence T_μ^η is $\|.\|_\infty$- contraction. □

This Lemma shows that T_μ^η is an approximation to T_μ and smaller values of η ensure that the fixed points of T_μ^η and T_μ are close.

5 About the Point of Convergence

The algorithm converges to the point θ^* such that $b - A\theta^* = 0$, from the analysis of Section 4. Now

$$\begin{aligned}
& b - A\theta^* = 0 \\
\implies & \Phi^T D_\mu \left((1+\eta) I - \gamma P_\pi\right) \Phi\theta^* = \Phi^T D_\mu r_\pi \\
\implies & \Phi^T D_\mu \left(I - \frac{\gamma}{1+\eta} P_\pi\right) \Phi\theta^* = \Phi^T D_\mu \frac{r_\pi}{1+\eta} \\
\implies & \Phi^T D_\mu \Phi\theta^* = \Phi^T D_\mu \left(\frac{r_\pi}{1+\eta} + \frac{\gamma}{1+\eta} P_\pi \Phi\theta^*\right) \\
\implies & \Phi\theta^* = \Phi(\Phi^T D_\mu \Phi)^{-1} \Phi^T D_\mu \left(\frac{r_\pi}{1+\eta} + \frac{\gamma}{1+\eta} P_\pi \Phi\theta^*\right) \\
\implies & \Phi\theta^* = \Pi_{D_\mu} T_\pi^\eta \Phi\theta^*,
\end{aligned}$$

where $\Pi_{D_\mu} = \Phi(\Phi^T D_\mu \Phi)^{-1} \Phi^T D_\mu$ is the projection operator that projects any $V \in \mathbb{R}^{|S|}$ to the subspace $\{\Phi r | r \in \mathbb{R}^K\}$ with respect to the norm $\|.\|_{D_\mu}$. Hence we observe that, similar to on-line on-policy TD(0), our on-line off-policy variant of TD(0) also converges to the fixed point of the projected perturbed Bellman operator. The projected perturbed Bellman operator in this case is $\Pi_{D_\mu} T_\pi^\eta$.

Remark 3 *Note that, in the case of $\Phi = I$ and $\eta = 0$, the solution of our algorithm $\theta^* = \Phi\theta^* = V_\pi$.*

Remark 4 *Note that the bound derived for η in Theorem 2 is a sufficient but not a necessary condition for the convergence of our proposed algorithm. If the value of η is large, the algorithm converges but to a poorly approximated solution. Therefore, in experiments, we select the value of η that is large enough to ensure convergence and small enough to ensure that the approximation is close.*

6 Experiments and Results

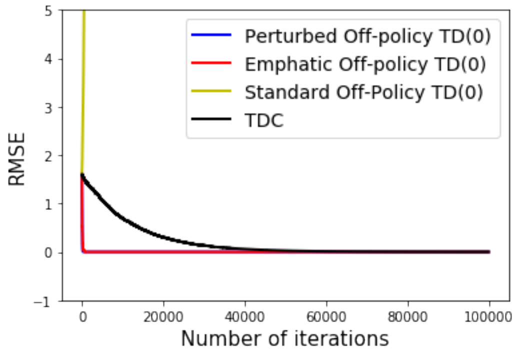


Figure 1: Performance of algorithms on "$\theta \rightarrow 2\theta$". RMSE is the value averaged across 10 independent runs.

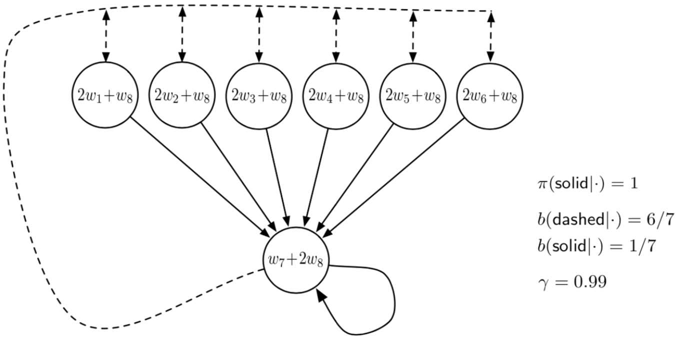
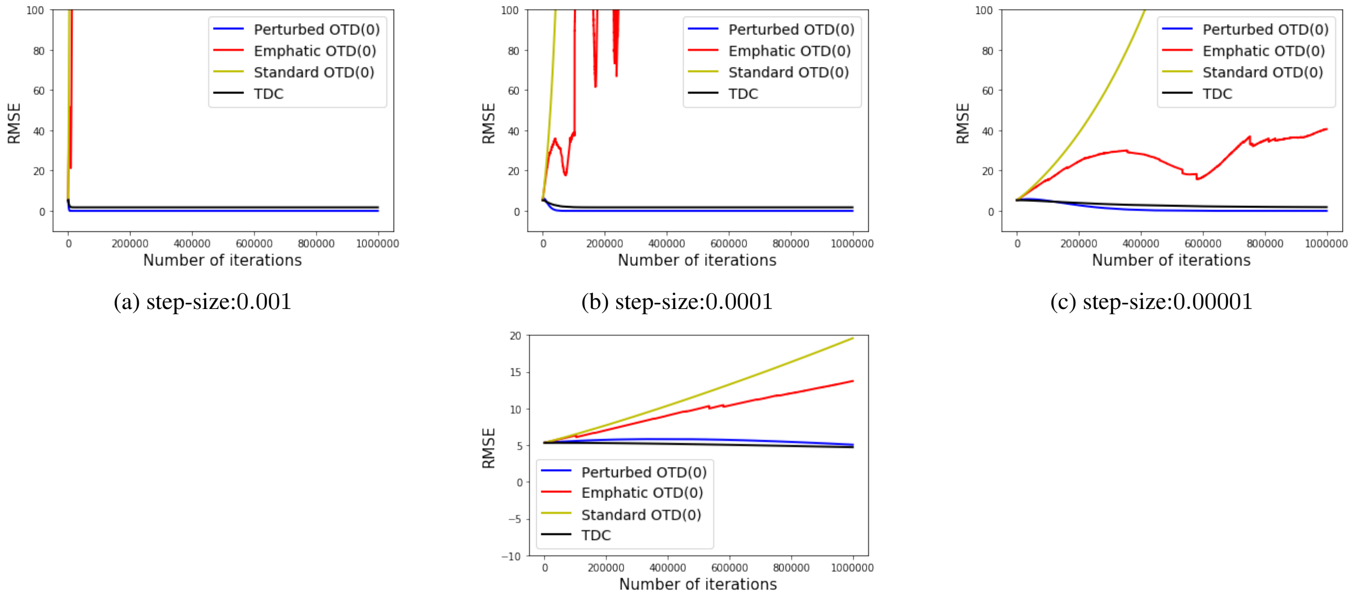


Figure 2: Baird's Counterexample. Figure taken from [25].



(a) step-size:0.001

(b) step-size:0.0001

(c) step-size:0.00001

(d) step-size:0.000001

Figure 3: Performance of algorithms on “Baird’s Counter-example”. RMSE is the value averaged across 10 runs. For the TDC algorithm, $\beta = 10\times$ step-size (Figure 5 of [18]).

In this section, we describe the performance of our proposed algorithm on three tasks. We first perform experiments on two benchmark counter-examples for off-policy divergence. Finally, we analyze the performance of our algorithm on a 3-state MDP example [2]. The evaluation metric considered is Root Mean Square Error (RMSE) defined as:

$$RMSE(\theta) = \sqrt{\sum_{s \in S} d_\mu(s)(V_\pi(s) - \widehat{V}_\theta(s))^2}, \qquad (10)$$

where θ is the parameter that is used to approximate the value function, d_μ is the stationary distribution of the Markov chain under the behavior policy μ, V_π is the exact value function of the target policy π and $\widehat{V}_\theta$ is the approximate value function that is estimated. For comparison purposes, we also implement Emphatic TD (ETD(0)) algorithm [19] and a gradient-family algorithm, linear TD with gradient correction (TDC) [18]. We perform 10 independent runs and present the average of RMSE obtained on all the three experiments.

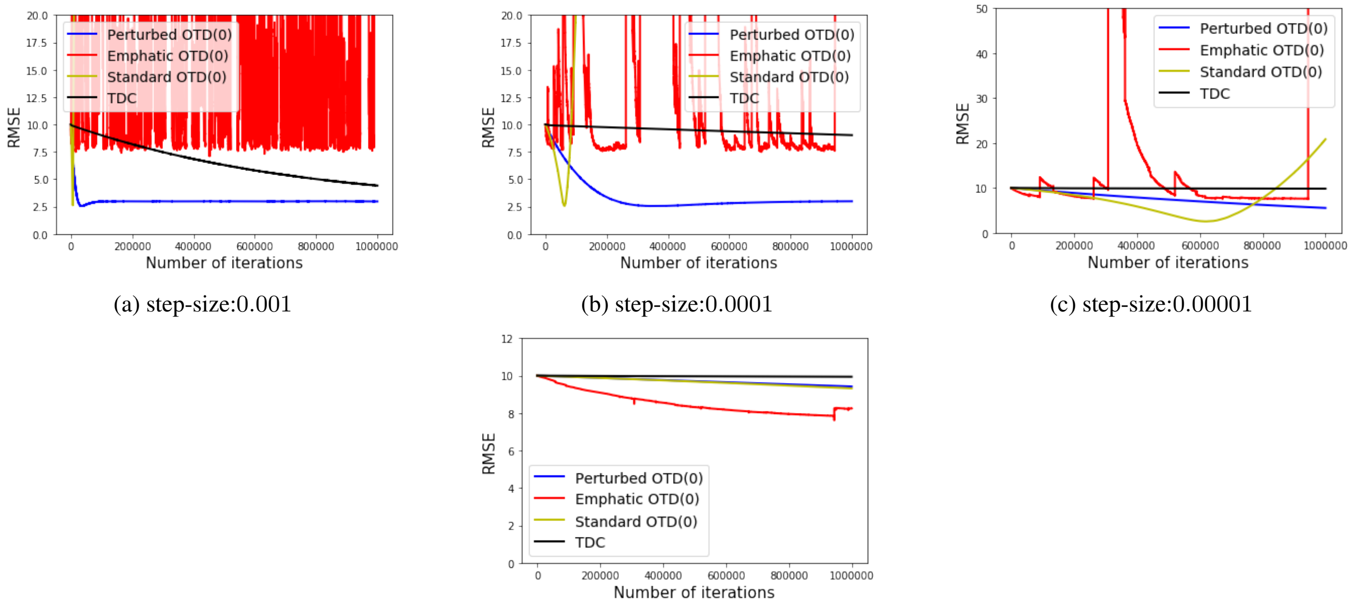
First, we consider the “$\theta \to 2\theta$” example ([22], Section 3 of [19]). In this example, there are two states - 1 and 2 and two actions - ‘left’ and ‘right’. Left action in state 1 results in state 1, while right action results in state 2. Similarly, right action in state 2 results in state 2 and left action results in state 1. The target policy is to take right in both the states, whereas the behavior policy is to take left and right actions with equal probability in both the states. The value function is linearly approximated with one feature. The feature of state 1 is 1 and that of state 2 is 2. The discount factor is taken to be 0.9. The update parameter θ is initialized to 1 and the η for our algorithm is taken to be 1. The step-size for the algorithms is held constant at 0.01. In Figure 1, we show the performance of algorithms over 100000 iterations. We can see that the standard off-policy TD(0) diverges whereas the other three algorithms including our proposed perturbed off-policy TD(0) converge to a point where the RMSE is zero.

Next, we consider the “7-star” example, first proposed in [1]. This is completely described in Figure 2 [25]. There are 7 states represented as circles. The expression inside the circle i represents the linear approximation of the state i. The policy π in Figure 2 represents the target policy and b represents the behavior policy. We run all the algorithms, i.e., standard off-policy TD(0), Emphatic off-policy TD(0), TDC and our algorithm, Perturbed off-policy TD(0) for 1000000 iterations. In Figure 3, we present the results of all the algorithms obtained by following different step-sizes. From Figures 3a,3b,3c, we can see that our perturbed off-policy TD converges to the exact solution while the Emphatic TD(0) appears to oscillate. Moreover, it is known that standard off-policy TD(0) diverges for this example, which can also be observed from Figures. In Figure 3d, the step-size considered is 0.000001 (which is very small) and therefore it may take a very large number of iterations to observe the behavior of the algorithms.

Finally, we construct an MDP as follows. There are 3 states and 2 actions - ‘left’ and ‘right’ possible in each state. The ‘left’ action in states 1 and 2 leads to state 1. And the ‘right’ action in states 2 and 3 leads to state 3. Finally ‘left’ action in state 3 leads to state 2. The single-stage rewards in all transitions are taken to be 1 and the discount factor is 0.9. The target policy $\pi = [[0, 1], [0.5, 0.5], [1, 0]]$ and the behavior policy $\mu = [[0.9, 0.1], [0.5, 0.5], [0.1, 0.9]]$ (where the first component represents the probability to take ‘left’ and the second component represents the probability to take ‘right’). The feature vectors of the three states are $[1, 0]$, $[1, 1]$, $[0, 1]$ respectively. We run all the algorithms for 1000000 iterations. Similar to our previous experiment, we present the results of all the algorithms obtained by following different step-sizes in Figure 4. From Figures 4a,4b,4c we can see that our proposed perturbed off-policy TD(0) converges. For this experiment, the best possible RMSE is 2.548 and our proposed algorithm (refer Figure 4b) achieves 2.97.

In the experimental setting above, the value of η is set to 0.5. In Figure 5, we run our algorithm with two other values of $\eta = 0.4$ and

[2] The implementation codes for our experiments is available at: `https://github.com/raghudiddigi/Off-Policy-Convergent-Algorithm`



(a) step-size:0.001

(b) step-size:0.0001

(c) step-size:0.00001

(d) step-size:0.000001

Figure 4: Performance of algorithms on a 3-state MDP. RMSE is the value averaged across 10 runs.The best possible RMSE for this example is 2.548. For the TDC algorithm, $\beta = 10\times$ step-size.

0.6 respectively. We observe that, for $\eta = 0.4$, convergence is not achieved as this η correction is not enough. On the other hand, for $\eta = 0.6$, convergence is ensured. However, the converged solution is not close due to the over-correction. Hence, it is to be noted that an optimal value of η is desired for ensuring the convergence and near-optimal solution at the same time (recall that a higher value of η is enough to ensure the convergence alone).

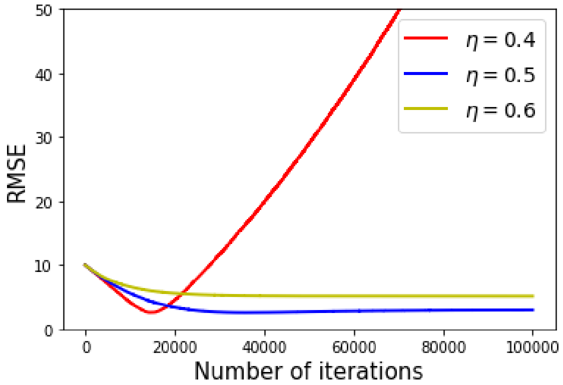


Figure 5: Performance of our proposed algorithm with three different η values.

Remark 5 *Note that the objective of the experiments here is to show that our proposed algorithm mitigates the divergence problem associated with the standard off-policy TD. If we choose a good value of η, it ensures that the algorithm converges to a solution close to the optimal solution. At this point, we do not make any claims about the quality of the converged solution of our proposed algorithm compared to that of the Emphatic TD(0) and TDC algorithms. Further empirical analysis is needed to conclusively make comparisons of the quality of converged solution using our algorithm with Emphatic TD(0), TDC as well as other off-policy algorithms in the literature.*

7 Conclusions and Future Work

In this work, we have proposed an off-policy TD algorithm for mitigating the divergence problem associated with the standard off-policy TD. Our proposed algorithm is simple in the sense that it trains only one set of parameters unlike GTD algorithms and doesn't use emphatic weights as in the ETD algorithm. It makes use of a perturbation parameter to ensure the convergence of the iterates. We proved that this addition of parameter makes the matrix A positive definite, which in turn ensures convergence. Finally, we empirically show the convergence on benchmark counter-examples for off-policy divergence.

As seen from the experiments, the choice of η is critical for our algorithm. The lower-bound that we have provided in our analysis is not tight and coming up with a tight bound is an interesting future direction. Our algorithm, in its current form, cannot be applied to compute the optimal policy (which is a control task). In future we would like to extend this algorithm to control tasks. Also, we would like to extend our algorithm to include eligibility traces and study its applications on real-world problems.

8 Acknowledgements

We thank the reviewers for their comments, which helped us to improve the paper. Raghuram Bharadwaj was supported by a fellowship grant from the Centre for Networked Intelligence (a Cisco CSR initiative) of the Indian Institute of Science, Bangalore. This work was supported by the Robert Bosch Centre for Cyber-Physical Systems, Indian Institute of Science, and a grant from the Department of Science and Technology, India. S.Bhatnagar was also supported by the J.C.Bose Fellowship.

References

[1] Leemon Baird, 'Residual algorithms: Reinforcement learning with function approximation', in *Machine Learning Proceedings 1995*, 30–37, Elsevier, (1995).

[2] Albert Benveniste, Michel Métivier, and Pierre Priouret, *Adaptive algorithms and stochastic approximations*, volume 22, Springer Science & Business Media, 2012.

[3] Dimitri P Bertsekas and John N Tsitsiklis, *Neuro-dynamic programming*, volume 5, Athena Scientific Belmont, MA, 1996.

[4] Steven J Bradtke and Andrew G Barto, 'Linear least-squares algorithms for temporal difference learning', *Machine learning*, **22**(1-3), 33–57, (1996).

[5] Carles Gelada and Marc G Bellemare, 'Off-policy deep reinforcement learning by bootstrapping the covariate shift', in *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3647–3655, (2019).

[6] Sina Ghiassian, Andrew Patterson, Martha White, Richard S Sutton, and Adam White, 'Online off-policy prediction', *arXiv preprint arXiv:1811.02597*, (2018).

[7] Sina Ghiassian, Banafsheh Rafiee, and Richard S Sutton, 'A first empirical study of emphatic temporal difference learning', *arXiv preprint arXiv:1705.04185*, (2017).

[8] GH Golub and CF Van Loan. Matrix computations, (Johns Hopkins University Press, Baltimore, 1996).

[9] Assaf Hallak and Shie Mannor, 'Consistent on-line off-policy evaluation', in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1372–1383. JMLR. org, (2017).

[10] Assaf Hallak, Aviv Tamar, Rémi Munos, and Shie Mannor, 'Generalized emphatic temporal difference learning: Bias-variance analysis', in *Thirtieth AAAI Conference on Artificial Intelligence*, (2016).

[11] Bo Liu, Sridhar Mahadevan, and Ji Liu, 'Regularized off-policy TD-learning', in *Advances in Neural Information Processing Systems*, pp. 836–844, (2012).

[12] Hamid R Maei, Csaba Szepesvári, Shalabh Bhatnagar, Doina Precup, David Silver, and Richard S Sutton, 'Convergent temporal-difference learning with arbitrary smooth function approximation', in *Advances in Neural Information Processing Systems*, pp. 1204–1212, (2009).

[13] Hamid Reza Maei and Richard S Sutton, 'GQ(λ): A general gradient algorithm for temporal-difference prediction learning with eligibility traces', in *3d Conference on Artificial General Intelligence (AGI-2010)*. Atlantis Press, (2010).

[14] Hamid Reza Maei, Csaba Szepesvári, Shalabh Bhatnagar, and Richard S Sutton, 'Toward off-policy learning control with function approximation.', in *ICML*, pp. 719–726, (2010).

[15] Doina Precup, Richard S Sutton, and Sanjoy Dasgupta, 'Off-policy temporal-difference learning with function approximation', in *ICML*, pp. 417–424, (2001).

[16] Jeffrey S Rosenthal, 'Convergence rates for Markov chains', *SIAM Review*, **37**(3), 387–405, (1995).

[17] Richard S Sutton and Andrew G Barto, *Reinforcement learning: An introduction*, MIT press, 2018.

[18] Richard S Sutton, Hamid Reza Maei, Doina Precup, Shalabh Bhatnagar, David Silver, Csaba Szepesvári, and Eric Wiewiora, 'Fast gradient-descent methods for temporal-difference learning with linear function approximation', in *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 993–1000. ACM, (2009).

[19] Richard S Sutton, A Rupam Mahmood, and Martha White, 'An emphatic approach to the problem of off-policy temporal-difference learning', *The Journal of Machine Learning Research*, **17**(1), 2603–2631, (2016).

[20] Richard S Sutton, Csaba Szepesvári, and Hamid Reza Maei, 'A convergent O(n) algorithm for off-policy temporal-difference learning with linear function approximation', *Advances in Neural Information Processing Systems*, **21**(21), 1609–1616, (2008).

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

[22] John N Tsitsiklis and Benjamin Van Roy, 'Feature-based methods for large scale dynamic programming', *Machine Learning*, **22**(1-3), 59–94, (1996).

[23] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg and Demis Hassabis , 'Human-level control through deep reinforcement learning', *Nature*, **518**(7540), 529, (2015).

[24] Huizhen Yu, 'On convergence of emphatic temporal-difference learning', in *Conference on Learning Theory*, pp. 1724–1751, (2015).

[25] Jeremy Zhang. Bairdexample. `https://github.com/MJeremy2017/Reinforcement-Learning-Implementation/tree/master/BairdExample`, November 2019.