A Variance Minimization Approach to Temporal-Difference Learning

Anonymous submission

Abstract

Under certain conditions, the larger the smallest eigenvalue of the key matrix of an algorithm, the faster the algorithm converges. By observation, most current objective functions aim to minimize error. Therefore, in this paper, we propose two new objective functions and derive three Variance Minimization (VM) algorithms, including VMTD, VMTDC and VMETD. A scalar parameter, ω, is introduced, to improve the performance of parametric Temporal-Difference (TD) learning algorithms. In the policy evaluation experiment, two-state, we analyze the convergence speed of these algorithms by calculating the minimum eigenvalue of the key matrices both on-policy and off-policy.In controlled experiments, the VM algorithms demonstrate superior performance.

Introduction

Reinforcement learning can be mainly divided into two categories: value-based reinforcement learning and policy gradient-based reinforcement learning. This paper focuses on temporal difference learning based on linear approximated valued functions. Its research is usually divided into two steps: the first step is to establish the convergence of the algorithm, and the second step is to accelerate the algorithm.

In terms of stability, Sutton (1988) established the convergence of on-policy TD(0), and Tsitsiklis and Van Roy (1997) established the convergence of on-policy TD(λ). However, "The deadly triad" consisting of off-policy learning, bootstrapping, and function approximation makes the stability a difficult problem (Sutton and Barto 2018). To solve this problem, convergent off-policy temporal difference learning algorithms are proposed, e.g., BR (Baird et al. 1995), GTD (Sutton, Maei, and Szepesvári 2008), GTD2 and TDC (Sutton et al. 2009), ETD (Sutton, Mahmood, and White 2016), and MRetrace (Chen et al. 2023).

In terms of acceleration, Hackman (2012) proposed Hybrid TD algorithm with on-policy matrix. Liu et al. (2015, 2016, 2018) proposed true stochastic algorithms, i.e., GTD-MP and GTD2-MP, from a convex-concave saddle-point formulation. Second-order methods are used to accelerate TD learning, e.g., Quasi Newton TD (Givchi and Palhang 2015) and accelerated TD (ATD) (Pan, White, and White 2017). Hallak et al. (2016) introduced an new parameter to reduce variance for ETD. Zhang and Whiteson (2022) proposed truncated ETD with a lower variance. Variance Reduced TD with direct variance reduction technique (Johnson and Zhang 2013) is proposed by (Korda and La 2015) and analysed by (Xu et al. 2019). How to further improve the convergence rates of reinforcement learning algorithms is currently still an open problem.

Algorithm stability is prominently reflected in the changes to the objective function, transitioning from mean squared errors (MSE) (Sutton and Barto 2018) to mean squared bellman errors (MSBE) (Baird et al. 1995), then to norm of the expected TD update (Sutton et al. 2009), and further to mean squared projected Bellman errors (MSPBE) (Sutton et al. 2009). On the other hand, algorithm acceleration is more centered around optimizing the iterative update formula of the algorithm itself without altering the objective function, thereby speeding up the convergence rate of the algorithm. The emergence of new optimization objective functions often leads to the development of novel algorithms. The introduction of new algorithms, in turn, tends to inspire researchers to explore methods for accelerating algorithms, leading to the iterative creation of increasingly superior algorithms.

The kernel loss function can be optimized using standard gradient-based methods, addressing the issue of double sampling in residual gradient algorithm (Feng, Li, and Liu 2019). It ensures convergence in both on-policy and off-policy scenarios. The logistic bellman error is convex and smooth in the action-value function parameters, with bounded gradients (Bas-Serrano et al. 2021). In contrast, the squared Bellman error is not convex in the action-value function parameters, and RL algorithms based on recursive optimization using it are known to be unstable.

It is necessary to propose a new objective function, but the mentioned objective functions above are all some form of error. Is minimizing error the only option for value-based reinforcement learning?

Based on this observation, we propose alternate objective functions instead of minimizing errors. We minimize Variance of Bellman Error (VBE) and Variance of Projected Bellman Error (VPBE) and derive Variance Minimization (VM) algorithms. These algorithms preserve the invariance of the optimal policy in the control environments, and significantly reduce the variance of gradient estimation, and thus hastening convergence.

The contributions of this paper are as follows:

- Introduction of novel objective functions, VBE and VPBE.
- Propose a on-policy VM algorithm and two off-policy VM algorithms.
- Proof of their convergence.
- The experiments demonstrate the superiority of the VM algorithms.

Background

Markov Decision Process

Reinforcement learning agent interacts with environment, observes state, takes sequential decision makings to influence environment, and obtains rewards. Consider an infinite-horizon discounted Markov Decision Process (MDP), defined by a tuple $\langle S, A, R, P, \gamma \rangle$, where $S = \{1, 2, \ldots, N\}$ is a finite set of states of the environment; A is a finite set of actions of the agent; $R : S \times A \times S \to \mathbb{R}$ is a bounded deterministic reward function; $P : S \times A \times S \to [0, 1]$ is the transition probability distribution; and $\gamma \in (0, 1)$ is the discount factor (Sutton and Barto 2018). Due to the requirements of online learning, value iteration based on sampling is considered in this paper. In each sampling, an experience (or transition) $\langle s, a, s', r \rangle$ is obtained.

A policy is a mapping $\pi : S \times A \to [0, 1]$. The goal of the agent is to find an optimal policy π^* to maximize the expectation of a discounted cumulative rewards in a long period. For each discrete time step $t = 0, 1, 2, 3, \ldots$, State value function $V^\pi(s)$ for a stationary policy π is defined as:

$$V^\pi(s) = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s].$$

Linear value function for state $s \in S$ is defined as:

$$V_\theta(s) := \theta^\top \phi(s) = \sum_{i=1}^{m} \theta_i \phi_i(s), \tag{1}$$

where $\theta := (\theta_1, \theta_2, \ldots, \theta_m)^\top \in \mathbb{R}^m$ is a parameter vector, $\phi := (\phi_1, \phi_2, \ldots, \phi_m)^\top \in \mathbb{R}^m$ is a feature function defined on state space S, and m is the feature size.

Tabular temporal difference (TD) learning (Sutton and Barto 2018) has been successfully applied to small-scale problems. To deal with the well-known curse of dimensionality of large scale MDPs, value function is usually approximated by a linear model (the focus of this paper), kernel methods, decision trees, or neural networks, etc.

On-policy and Off-policy

On-policy and off-policy algorithms are currently hot topics in research. The main difference between the two lies in the fact that in on-policy algorithms, the behavior policy μ and the target policy π are the same during the learning process. In off-policy algorithms, however, the behavior policy and the target policy are different. The algorithm uses data generated from the behavior policy to optimize the target policy, which leads to higher sample efficiency and complex stability issues.

From the theory of stochastic methods, the convergence point of linear TD algorithms, is a parameter vector, say $\boldsymbol{\theta}$, that satisfies

$$b - \mathbf{A}\theta \quad = \quad 0,$$

where $\mathbf{A} \in \mathbb{R}^{|S| \times m}$ and $b \in \mathbb{R}^m$. If the matrix $\mathbf{A}$ is positive definite, then the algorithm converges. The convergence rate of the algorithm is related to the matrix $\mathbf{A}$. The larger the minimum eigenvalue of $\mathbf{A}$, the faster the convergence rate. Next, we will compute the minimum eigenvalue of $\mathbf{A}$ for TD(0), TDC, and ETD in both on-policy and off-policy settings in a 2-state environment. First, we will introduce the environment setup for the 2-state case in both on-policy and off-policy settings.

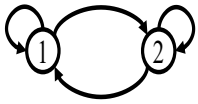


Figure 1: 2-state

The "1"→"2" problem has only two states. From each state, there are two actions, left and right, which take the agent to the left or right state. All rewards are zero. The feature $\boldsymbol{\Phi} = (1, 2)^\top$ are assigned to the left and the right state. The first policy takes the equal probability to left or right in both states, i.e., $\mathbf{P}_1 = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$. The second policy only selects action right in both states, i.e., $\mathbf{P}_2 = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$. The state distribution of the first policy is $d_1 = (0.5, 0.5)^\top$. The state distribution of the second policy is $d_1 = (0, 1)^\top$. The discount factor is $\gamma = 0.9$. In the on-policy setting, the behavior policy and the target policy are the same, so let $\mathbf{P}_\mu = \mathbf{P}_\pi = \mathbf{P}_1$. In the off-policy setting, let $\mathbf{P}_\mu = \mathbf{P}_1$ and $\mathbf{P}_\pi = \mathbf{P}_2$.

The key matrix $\mathbf{A}_{\text{on}}$ of on-policy TD(0) is

$$\mathbf{A}_{\text{on}} = \boldsymbol{\Phi}^\top \mathbf{D}_\pi (\mathbf{I} - \gamma \mathbf{P}_\pi) \boldsymbol{\Phi},$$

where $\boldsymbol{\Phi}$ is the $|S| \times m$ matrix with the $\phi(s)$ as its rows, and $\mathbf{D}_\pi$ is the $|S| \times |S|$ diagonal matrix with d_π on its diagonal. d_π is a vector, each component representing the steady-state distribution under policy π. $\mathbf{P}_\pi$ denote the $|S| \times |S|$ matrix of transition probabilities under π. And $\mathbf{P}_\pi^\top d_\pi = d_\pi$.

The key matrix $\mathbf{A}_{\text{off}}$ of off-policy TD(0) is

$$\mathbf{A}_{\text{off}} = \boldsymbol{\Phi}^\top \mathbf{D}_\mu (\mathbf{I} - \gamma \mathbf{P}_\pi) \boldsymbol{\Phi},$$

where $\mathbf{D}_\mu$ is the $|S| \times |S|$ diagonal matrix with d_μ on its diagonal. d_μ is a vector, each component representing the steady-state distribution under behavior policy μ.

In the off-policy 2-state, $\mathbf{A}_{\text{off}} = -0.2$, which means that off-policy TD(0) cannot stably converge, while , in the on-policy 2-state, $\mathbf{A}_{\text{on}} = 0.475$, which means that on-policy TD(0) can stably converge.

The key matrix $\mathbf{A}_{\text{TDC}} = \mathbf{A}_{\text{off}}^\top \mathbf{C}^{-1} \mathbf{A}_{\text{off}}$, where $\mathbf{C} = \mathbb{E}[\boldsymbol{\phi}\boldsymbol{\phi}^\top]$. In the 2-state counterexample, $\mathbf{A}_{\text{TDC}} = 0.016$, which means that TDC can stably converge.

Table 1: Minimum eigenvalues of various algorithms in the 2-state counterexample.

ALGORITHM	TD	TDC	ETD	VMTD	VMTDC	VMETD
ON-POLICY 2-STATE	0.475	0.09025	\	0.25	0.025	\
OFF-POLICY 2-STATE	-0.2	0.016	3.4	0.25	0.025	1.15

The key matrix $\mathbf{A}_{\text{TDC}}$ of on-policy TDC is

$$\mathbf{A}_{\text{TDC}} = \mathbf{A}_{\text{on}}^\top \mathbf{C}^{-1} \mathbf{A}_{\text{on}}.$$

The key matrix $\mathbf{A}_{\text{TDC}}$ of off-policy TDC is

$$\mathbf{A}_{\text{TDC}} = \mathbf{A}_{\text{off}}^\top \mathbf{C}^{-1} \mathbf{A}_{\text{off}}.$$

$\mathbf{A}_{\text{TDC}} = 0.016$ in the off-policy 2-state and $\mathbf{A}_{\text{TDC}} = 0.09025$ in the on-policy 2-state, which means that TDC can stably converge in two settings.

To address the issue of the key matrix $\mathbf{A}_{\text{off}}$ in off-policy TD(0) being non-positive definite, a scalar variable, F_t, is introduced to obtain the off-policy TD(0) algorithm, which ensures convergence under off-policy conditions.

The key matrix $\mathbf{A}_{\text{ETD}}$ is

$$\mathbf{A}_{\text{ETD}} = \mathbf{\Phi}^\top \mathbf{F} (\mathbf{I} - \gamma \mathbf{P}_\pi) \mathbf{\Phi},$$

where $\mathbf{F}$ is a diagonal matrix with diagonal elements $f(s) \doteq d_\mu(s) \lim_{t\to\infty} \mathbb{E}_\mu[F_t | S_t = s]$, which we assume exists. The vector $\mathbf{f} \in \mathbb{R}^N$ with components $[\mathbf{f}]_s \doteq f(s)$ can be written as

$$\begin{aligned} \mathbf{f} &= \mathbf{d}_\mu + \gamma \mathbf{P}_\pi^\top \mathbf{d}_\mu + (\gamma \mathbf{P}_\pi^\top)^2 \mathbf{d}_\mu + \dots \\ &= (\mathbf{I} - \gamma \mathbf{P}_\pi^\top)^{-1} \mathbf{d}_\mu. \end{aligned}$$

In the off-policy 2-state, $\mathbf{A}_{\text{ETD}} = 3.4$, which means that ETD can stably converge.

Table 1 shows Minimum eigenvalues of various algorithms in the 2-state counterexample.

In the on-policy 2-state environment, the minimum eigenvalue of the key matrix for TDC is greater than that of TD(0), indicating that TDC converges faster than TD(0) in this environment. In the off-policy 2-state environment, the minimum eigenvalue of the key matrix for ETD is the largest, suggesting that ETD has the fastest convergence rate.

Minimum eigenvalue larger, algorithm's convergence faster. To derive an algorithm with a larger minimum eigenvalue for matrix $\mathbf{A}$, it is necessary to propose new objective functions. The mentioned objective functions in the Introduction are all forms of error. Is minimizing error the only option for value-based reinforcement learning? Based on this observation, we propose alternative objective functions instead of minimizing errors.

Variance Minimization Algorithms

This section will introduce two new objective functions and three new algorithms, including one on-policy algorithm and two off-policy algorithms, and calculate the minimum eigenvalue of $\mathbf{A}$ for each of the three algorithms under on-policy and off-policy in a 2-state environment, thereby comparing the convergence speed of the three algorithms.

Variance Minimization TD Learning: VMTD

For on-policy learning, a novel objective function, Variance of Bellman Error (VBE), is proposed as follows:

$$\begin{aligned} \arg\min_\theta \text{VBE}(\theta) &= \arg\min_\theta \mathbb{E}[(\mathbb{E}[\delta_t|S_t] - \mathbb{E}[\mathbb{E}[\delta_t|S_t]])^2] \\ &= \arg\min_{\theta,\omega} \mathbb{E}[(\mathbb{E}[\delta_t|S_t] - \omega)^2] \end{aligned} \tag{2}$$

where δ_t is the TD error as follows:

$$\delta_t = r_{t+1} + \gamma \theta_t^\top \phi_{t+1} - \theta_t^\top \phi_t. \tag{3}$$

Clearly, it is no longer to minimize Bellman errors.

First, the parameter ω is derived directly based on stochastic gradient descent:

$$\omega_{t+1} \leftarrow \omega_t + \beta_t(\delta_t - \omega_t), \tag{4}$$

Then, based on stochastic semi-gradient descent, the update of the parameter θ is as follows:

$$\theta_{t+1} \leftarrow \theta_t + \alpha_t(\delta_t - \omega_t)\phi_t. \tag{5}$$

The semi-gradient of the (2) with respect to θ is

$$\begin{aligned} & -\tfrac{1}{2}\nabla \text{VBE}(\theta) \\ =\ & \mathbb{E}[(\mathbb{E}[\delta_t|S_t] - \mathbb{E}[\mathbb{E}[\delta_t|S_t]])(\phi_t - \mathbb{E}[\phi_t])] \\ =\ & \mathbb{E}[\delta_t \phi_t] - \mathbb{E}[\delta_t]\mathbb{E}[\phi_t], \end{aligned}$$

The key matrix $\mathbf{A}_{\text{VMTD}}$ and b_{VMTD} of on-policy VMTD is

$$\begin{aligned} & \mathbf{A}_{\text{VMTD}} \\ =\ & \mathbb{E}[(\phi - \gamma\phi')\phi^\top] - \mathbb{E}[\phi - \gamma\phi']\mathbb{E}[\phi^\top] \\ =\ & \textstyle\sum_s d_\pi(s)\phi(s)\Big(\phi(s) - \gamma \sum_{s'} [\mathbf{P}_\pi]_{ss'}\phi(s')\Big)^\top \\ & - \textstyle\sum_s d_\pi(s)\phi(s) \cdot \sum_s d_\pi(s)\Big(\phi(s) - \gamma \sum_{s'} [\mathbf{P}_\pi]_{ss'}\phi(s')\Big)^\top \\ =\ & \mathbf{\Phi}^\top \mathbf{D}_\mu (\mathbf{I} - \gamma \mathbf{P}_\pi)\mathbf{\Phi} - \mathbf{\Phi}^\top d_\pi d_\pi^\top (\mathbf{I} - \gamma \mathbf{P}_\pi)\mathbf{\Phi} \\ =\ & \mathbf{\Phi}^\top (\mathbf{D}_\pi - d_\pi d_\pi^\top)(\mathbf{I} - \gamma \mathbf{P}_\pi)\mathbf{\Phi}, \end{aligned}$$

$$\begin{aligned} & b_{\text{VMTD}} \\ =\ & \mathbb{E}(r - \mathbb{E}[r])\phi \\ =\ & \mathbb{E}[r\phi] - \mathbb{E}[r]\mathbb{E}[\phi] \\ =\ & \mathbf{\Phi}^\top (\mathbf{D}_\pi - d_\pi d_\pi^\top) r_\pi. \end{aligned}$$

It can be easily obtained that The key matrix $\mathbf{A}_{\text{VMTD}}$ and b_{VMTD} of off-policy VMTD are, respectively,

$$\mathbf{A}_{\text{VMTD}} = \mathbf{\Phi}^\top (\mathbf{D}_\mu - d_\mu d_\mu^\top)(\mathbf{I} - \gamma \mathbf{P}_\pi)\mathbf{\Phi},$$

$$b_{\text{VMTD}} = \mathbf{\Phi}^\top (\mathbf{D}_\mu - d_\mu d_\mu^\top) r_\pi,$$

In the on-policy 2-state environment, the minimum eigenvalue of the key matrix for VMTD is greater than that of on-policy TDC and smaller than that of on-policy TD(0), indicating that VMTD converges faster than TDC and slower than TD(0) in this environment. In the off-policy 2-state environment, the minimum eigenvalue of the key matrix for VMTD is greater than 0, suggesting that VMTD can converge stably.

Variance Minimization TDC Learning: VMTDC

For off-policy learning, we propose a new objective function, called Variance of Projected Bellman error (VPBE), and the corresponding algorithm is called VMTDC.

$$\begin{aligned}&\text{VPBE}(\boldsymbol{\theta})\\&= \mathbb{E}[(\delta - \mathbb{E}[\delta])\boldsymbol{\phi}]^\top \mathbb{E}[\boldsymbol{\phi}\boldsymbol{\phi}^\top]^{-1}\mathbb{E}[(\delta - \mathbb{E}[\delta])\boldsymbol{\phi}] \quad (6)\\&= \mathbb{E}[(\delta - \omega)\boldsymbol{\phi}]^\top \mathbb{E}[\boldsymbol{\phi}\boldsymbol{\phi}^\top]^{-1}\mathbb{E}[(\delta - \omega)\boldsymbol{\phi}], \quad (7)\end{aligned}$$

where ω is used to approximate $\mathbb{E}[\delta]$, i.e., $\omega \doteq \mathbb{E}[\delta]$.

The gradient of the (6) with respect to θ is

$$\begin{aligned}-\tfrac{1}{2}\nabla \text{VPBE}(\theta) &= -\mathbb{E}\Big[\Big((\gamma\phi' - \phi) - \mathbb{E}[(\gamma\phi' - \phi)]\Big)\phi^\top\Big]\\&\quad \mathbb{E}[\phi\phi^\top]^{-1}\mathbb{E}[(\delta - \mathbb{E}[\delta])\phi]\\&= \mathbb{E}\Big[\Big((\phi - \gamma\phi') - \mathbb{E}[(\phi - \gamma\phi')]\Big)\phi^\top\Big]\\&\quad \mathbb{E}[\phi\phi^\top]^{-1}\\&\quad \mathbb{E}\Big[\Big(r + \gamma\phi'^\top\theta - \phi^\top\theta\\&\qquad -\mathbb{E}[r + \gamma\phi'^\top\theta - \phi^\top\theta]\Big)\phi\Big].\end{aligned}$$

It can be easily obtained that The key matrix $\mathbf{A}_{\text{VMTDC}}$ and b_{VMTDC} of VMTDC are, respectively,

$$\mathbf{A}_{\text{VMTDC}} = \mathbf{A}_{\text{VMTD}}^\top \mathbf{C}^{-1}\mathbf{A}_{\text{VMTD}},$$

$$b_{\text{VMTDC}} = \mathbf{A}_{\text{VMTD}}^\top \mathbf{C}^{-1} b_{\text{VMTD}},$$

where, for on-policy, $\mathbf{A}_{\text{VMTD}} = \mathbf{\Phi}^\top(\mathbf{D}_\pi - d_\pi d_\pi^\top)(\mathbf{I} - \gamma\mathbf{P}_\pi)\mathbf{\Phi}$ and $b_{\text{VMTD}} = \mathbf{\Phi}^\top(\mathbf{D}_\pi - d_\pi d_\pi^\top)r_\pi$ and, for off-policy, $\mathbf{A}_{\text{VMTD}} = \mathbf{\Phi}^\top(\mathbf{D}_\mu - d_\mu d_\mu^\top)(\mathbf{I} - \gamma\mathbf{P}_\pi)\mathbf{\Phi}$ and $b_{\text{VMTD}} = \mathbf{\Phi}^\top(\mathbf{D}_\mu - d_\mu d_\mu^\top)r_\pi$.

In the process of computing the gradient of the (7) with respect to θ, ω is treated as a constant. So, the derivation process of the VMTDC algorithm is the same as that of the TDC algorithm, the only difference is that the original δ is replaced by $\delta - \omega$. Therefore, we can easily get the updated formula of VMTDC, as follows:

$$\boldsymbol{\theta}_{k+1} \leftarrow \boldsymbol{\theta}_k + \alpha_k[(\delta_k - \omega_k)\boldsymbol{\phi}_k - \gamma\boldsymbol{\phi}_{k+1}(\boldsymbol{\phi}_k^\top \boldsymbol{u}_k)], \quad (8)$$

$$\boldsymbol{u}_{k+1} \leftarrow \boldsymbol{u}_k + \zeta_k[\delta_k - \omega_k - \boldsymbol{\phi}_k^\top \boldsymbol{u}_k]\boldsymbol{\phi}_k, \quad (9)$$

and

$$\omega_{k+1} \leftarrow \omega_k + \beta_k(\delta_k - \omega_k). \quad (10)$$

The VMTDC algorithm (8) is derived to work with a given set of sub-samples—in the form of triples (S_k, R_k, S'_k) that match transitions from both the behavior and target policies.

In the on-policy 2-state environment, the minimum eigenvalue of the key matrix for VMTDC is smaller than that of TD(0), TDC and VMTD indicating that VMTDC converges slower than them in this on-policy. In the off-policy 2-state environment, the minimum eigenvalue of the key matrix for VMTD is greater than TDC, suggesting that VMTDC converges faster than them in off-policy environment.

Variance Minimization ETD Learning: VMETD

Based on the off-policy TD algorithm, a scalar, F, is introduced to obtain the ETD algorithm, which ensures convergence under off-policy conditions. This paper further introduces a scalar, ω, based on the ETD algorithm to obtain VMETD. VMETD by the following update:

$$F_t \leftarrow \gamma\rho_{t-1}F_{t-1} + 1, \quad (11)$$

$$\theta_{t+1} \leftarrow \theta_t + \alpha_t(F_t\rho_t\delta_t - \omega_t)\phi_t, \quad (12)$$

$$\omega_{t+1} \leftarrow \omega_t + \beta_t(F_t\rho_t\delta_t - \omega_t), \quad (13)$$

where ω is used to estimate $\mathbb{E}[F\rho\delta]$, i.e., $\omega \doteq \mathbb{E}[F\rho\delta]$.

(12) can be rewritten as

$$\begin{aligned}\theta_{t+1} &\leftarrow \theta_t + \alpha_t(F_t\rho_t\delta_t - \omega_t)\phi_t - \alpha_t\omega_{t+1}\phi_t\\&= \theta_t + \alpha_t(F_t\rho_t\delta_t - \mathbb{E}_\mu[F_t\rho_t\delta_t|\theta_t])\phi_t\\&= \theta_t + \alpha_t F_t\rho_t(r_{t+1} + \gamma\theta_t^\top\phi_{t+1} - \theta_t^\top\phi_t)\phi_t\\&\quad -\alpha_t\mathbb{E}_\mu[F_t\rho_t\delta_t]\phi_t\\&= \theta_t + \alpha_t\{\underbrace{(F_t\rho_t r_{t+1} - \mathbb{E}_\mu[F_t\rho_t r_{t+1}])\phi_t}_{b_{\text{VMETD},t}}\\&\quad \underbrace{-(F_t\rho_t\phi_t(\phi_t - \gamma\phi_{t+1})^\top - \phi_t\mathbb{E}_\mu[F_t\rho_t(\phi_t - \gamma\phi_{t+1})]^\top)}_{\mathbf{A}_{\text{VMETD},t}}\theta_t\}.\end{aligned}$$

Therefore,

$$\begin{aligned}&\mathbf{A}_{\text{VMETD}}\\&= \lim_{t\to\infty}\mathbb{E}[\mathbf{A}_{\text{VMETD},t}]\\&= \lim_{t\to\infty}\mathbb{E}_\mu[F_t\rho_t\phi_t(\phi_t - \gamma\phi_{t+1})^\top]\\&\quad - \lim_{t\to\infty}\mathbb{E}_\mu[\phi_t]\mathbb{E}_\mu[F_t\rho_t(\phi_t - \gamma\phi_{t+1})]^\top\\&= \lim_{t\to\infty}\mathbb{E}_\mu[\phi_t F_t\rho_t(\phi_t - \gamma\phi_{t+1})^\top]\\&\quad - \lim_{t\to\infty}\mathbb{E}_\mu[\phi_t]\lim_{t\to\infty}\mathbb{E}_\mu[F_t\rho_t(\phi_t - \gamma\phi_{t+1})]^\top\\&= \textstyle\sum_s d_\mu(s)\lim_{t\to\infty}\mathbb{E}_\mu[F_t|S_t = s]\mathbb{E}_\mu[\rho_t\phi_t(\phi_t - \gamma\phi_{t+1})^\top|S_t = s]\\&\quad - \textstyle\sum_s d_\mu(s)\phi(s)\sum_s d_\mu(s)\lim_{t\to\infty}\mathbb{E}_\mu[F_t|S_t = s]\\&\quad \mathbb{E}_\mu[\rho_t(\phi_t - \gamma\phi_{t+1})^\top|S_t = s]\\&= \textstyle\sum_s f(s)\mathbb{E}_\pi[\phi_t(\phi_t - \gamma\phi_{t+1})^\top|S_t = s]\\&\quad - \textstyle\sum_s d_\mu(s)\phi(s)\sum_s f(s)\mathbb{E}_\pi[(\phi_t - \gamma\phi_{t+1})^\top|S_t = s]\\&= \textstyle\sum_s f(s)\boldsymbol{\phi}(s)(\boldsymbol{\phi}(s) - \gamma\sum_{s'}[\mathbf{P}_\pi]_{ss'}\boldsymbol{\phi}(s'))^\top\\&\quad - \textstyle\sum_s d_\mu(s)\phi(s) * \sum_s f(s)(\phi(s) - \gamma\sum_{s'}[\mathbf{P}_\pi]_{ss'}\phi(s'))^\top\\&= \mathbf{\Phi}^\top\mathbf{F}(\mathbf{I} - \gamma\mathbf{P}_\pi)\mathbf{\Phi} - \mathbf{\Phi}^\top d_\mu f^\top(\mathbf{I} - \gamma\mathbf{P}_\mu)\mathbf{\Phi}\\&= \mathbf{\Phi}^\top(\mathbf{F} - d_\mu f^\top)(\mathbf{I} - \gamma\mathbf{P}_\pi)\mathbf{\Phi}\\&= \mathbf{\Phi}^\top(\mathbf{F}(\mathbf{I} - \gamma\mathbf{P}_\pi) - d_\mu f^\top(\mathbf{I} - \gamma\mathbf{P}_\pi))\mathbf{\Phi}\\&= \mathbf{\Phi}^\top(\mathbf{F}(\mathbf{I} - \gamma\mathbf{P}_\pi) - d_\mu d_\mu^\top)\mathbf{\Phi},\end{aligned}$$

$$\begin{aligned}&b_{\text{VMETD}}\\&= \lim_{t\to\infty}\mathbb{E}[b_{\text{VMETD},t}]\\&= \lim_{t\to\infty}\mathbb{E}_\mu[F_t\rho_t R_{t+1}\phi_t]\\&\quad - \lim_{t\to\infty}\mathbb{E}_\mu[\phi_t]\mathbb{E}_\mu[F_t\rho_k R_{k+1}]\\&= \lim_{t\to\infty}\mathbb{E}_\mu[\phi_t F_t\rho_t r_{t+1}]\\&\quad - \lim_{t\to\infty}\mathbb{E}_\mu[\phi_t]\mathbb{E}_\mu[\phi_t]\mathbb{E}_\mu[F_t\rho_t r_{t+1}]\\&= \lim_{t\to\infty}\mathbb{E}_\mu[\phi_t F_t\rho_t r_{t+1}]\\&\quad - \lim_{t\to\infty}\mathbb{E}_\mu[\phi_t]\lim_{t\to\infty}\mathbb{E}_\mu[F_t\rho_t r_{t+1}]\\&= \textstyle\sum_s f(s)\phi(s)r_\pi - \sum_s d_\mu(s)\phi(s) * \sum_s f(s)r_\pi\\&= \mathbf{\Phi}^\top(\mathbf{F} - d_\mu f^\top)r_\pi.\end{aligned}$$

In the off-policy 2-state environment, the minimum eigenvalue of the key matrix for VMETD is greater than that of TD(0), TDC and VMTD and smaller than that of ETD, indicating that VMTDC converges faster than TD(0), TDC and VMTD and slower than ETD in this off-policy. However, subsequent experiments showed that the VMETD algorithm converges more smoothly and performs best in controlled experiments.

Theoretical Analysis

This section primarily focuses on proving the convergence of VMTD, VMTDC, and VMETD.

Theorem 1. *(Convergence of VMTD). In the case of on-policy learning, consider the iterations (4) and (5) with (3) of VMTD. Let the step-size sequences α_k and β_k, $k \geq 0$ satisfy in this case $\alpha_k, \beta_k > 0$, for all k, $\sum_{k=0}^{\infty} \alpha_k = \sum_{k=0}^{\infty} \beta_k = \infty$, $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$, $\sum_{k=0}^{\infty} \beta_k^2 < \infty$, and $\alpha_k = o(\beta_k)$. Assume that (ϕ_k, r_k, ϕ_k') is an i.i.d. sequence with uniformly bounded second moments, where ϕ_k and ϕ_k' are sampled from the same Markov chain. Let $\boldsymbol{A} = \mathrm{Cov}(\phi, \phi - \gamma\phi')$, $b = \mathrm{Cov}(r, \phi)$. Assume that matrix $\boldsymbol{A}$ is non-singular. Then the parameter vector θ_k converges with probability one to $\boldsymbol{A}^{-1}b$.*

Proof. The proof is based on Borkar's Theorem for general stochastic approximation recursions with two time scales (Borkar 1997).

A sketch proof is given as follows. In the fast time scale, the parameter w converges to $\mathbb{E}[\delta|\theta_k]$. In the slow time scale, the associated ODE is

$$\vec{h}(\theta(t)) = -\mathbf{A}\theta(t) + b.$$

$$\begin{aligned} A &= \mathrm{Cov}(\phi, \phi - \gamma\phi') \\ &= \tfrac{\mathrm{Cov}(\phi,\phi) + \mathrm{Cov}(\phi - \gamma\phi', \phi - \gamma\phi') - \mathrm{Cov}(\gamma\phi', \gamma\phi')}{2} \\ &= \tfrac{\mathrm{Cov}(\phi,\phi) + \mathrm{Cov}(\phi - \gamma\phi', \phi - \gamma\phi') - \gamma^2\mathrm{Cov}(\phi', \phi')}{2} \\ &= \tfrac{(1-\gamma^2)\mathrm{Cov}(\phi,\phi) + \mathrm{Cov}(\phi - \gamma\phi', \phi - \gamma\phi')}{2}, \end{aligned} \tag{14}$$

where we eventually used $\mathrm{Cov}(\phi', \phi') = \mathrm{Cov}(\phi, \phi)$ [1]. Note that the covariance matrix $\mathrm{Cov}(\phi, \phi)$ and $\mathrm{Cov}(\phi - \gamma\phi', \phi - \gamma\phi')$ are semi-positive definite. Then, the matrix $\mathbf{A}$ is semi-positive definite because $\mathbf{A}$ is linearly combined by two positive-weighted semi-positive definite matrice (14). Furthermore, $\mathbf{A}$ is nonsingular due to the assumption. Hence, the matrix $\mathbf{A}$ is positive definite. And, the parameter θ converges to $\mathbf{A}^{-1}b$. □

Please refer to the appendix for VMTD's detailed proof process.

Theorem 2. *(Convergence of VMTDC). In the case of off-policy learning, consider the iterations (10), (9) and (8) of VMTDC. Let the step-size sequences α_k, ζ_k and β_k, $k \geq 0$ satisfy in this case $\alpha_k, \zeta_k, \beta_k > 0$, for all k, $\sum_{k=0}^{\infty} \alpha_k = \sum_{k=0}^{\infty} \beta_k = \sum_{k=0}^{\infty} \zeta_k = \infty$, $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$, $\sum_{k=0}^{\infty} \zeta_k^2 < \infty$, $\sum_{k=0}^{\infty} \beta_k^2 < \infty$, and $\alpha_k = o(\zeta_k)$, $\zeta_k = o(\beta_k)$. Assume that (ϕ_k, r_k, ϕ_k') is an i.i.d. sequence with uniformly bounded second moments. Let $\boldsymbol{A} = \mathrm{Cov}(\phi, \phi - \gamma\phi')$, $b = \mathrm{Cov}(r, \phi)$, and $\boldsymbol{C} = \mathbb{E}[\phi\phi^\top]$. Assume that $\boldsymbol{A}$ and $\boldsymbol{C}$ are non-singular matrices. Then the parameter vector θ_k converges with probability one to $\boldsymbol{A}^{-1}b$.*

Proof. The proof is similar to that given by (Sutton et al. 2009) for TDC, but it is based on multi-time-scale stochastic approximation.

[1] The covariance matrix $\mathrm{Cov}(\phi', \phi')$ is equal to the covariance matrix $\mathrm{Cov}(\phi, \phi)$ if the initial state is re-reachable or initialized randomly in a Markov chain for on-policy update.

A sketch proof is given as follows. In the fastest time scale, the parameter w converges to $\mathbb{E}[\delta|u_k, \theta_k]$. In the second fast time scale, the parameter u converges to $\mathbf{C}^{-1}\mathbb{E}[(\delta - \mathbb{E}[\delta|\theta_k])\phi|\theta_k]$. In the slower time scale, the associated ODE is

$$\vec{h}(\theta(t)) = \mathbf{A}^\top\mathbf{C}^{-1}(-\mathbf{A}\theta(t) + b).$$

The matrix $\mathbf{A}^\top\mathbf{C}^{-1}\mathbf{A}$ is positive definite. Thus, the parameter θ converges to $\mathbf{A}^{-1}b$. □

Please refer to the appendix for VMTDC's detailed proof process.

Theorem 3. *(Convergence of VMETD). In the case of off-policy learning, consider the iterations (11), (13) and (12) of VMETD. Let the step-size sequences α_k and β_k, $k \geq 0$ satisfy in this case $\alpha_k, \beta_k > 0$, for all k, $\sum_{k=0}^{\infty} \alpha_k = \sum_{k=0}^{\infty} \beta_k = \infty$, $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$, $\sum_{k=0}^{\infty} \beta_k^2 < \infty$, and $\alpha_k = o(\beta_k)$. Assume that $(\boldsymbol{\phi}_k, r_k, \boldsymbol{\phi}_k')$ is an i.i.d. sequence with uniformly bounded second moments, where $\boldsymbol{\phi}_k$ and $\boldsymbol{\phi}_k'$ are sampled from the same Markov chain. Let $\boldsymbol{A_{VMETD}} = \boldsymbol{\Phi}^\top(\boldsymbol{F}(\boldsymbol{I} - \gamma\boldsymbol{P}_\pi) - d_\mu d_\mu^\top)\boldsymbol{\Phi}$, $b_{VMETD} = \boldsymbol{\Phi}^\top(\boldsymbol{F} - d_\mu f^\top)r_\pi$. Assume that matrix $\boldsymbol{A}$ is non-singular. Then the parameter vector θ_k converges with probability one to $\boldsymbol{A}_{\boldsymbol{VMETD}}^{-1}b_{\boldsymbol{VMETD}}$.*

Proof. The proof of VMETD's convergence is also based on Borkar's Theorem for general stochastic approximation recursions with two time scales (Borkar 1997).

A sketch proof is given as follows. In the fast time scale, the parameter ω converges to $\mathbb{E}_\mu[F\rho\delta|\theta_k]$. Recursion (12) is considered the slower timescale. If the key matrix $\mathbf{A}_{\text{VMETD}}$ is positive definite, then θ converges.

$$\begin{aligned} &(\mathbf{F}(\mathbf{I} - \gamma\mathbf{P}_\pi) - d_\mu d_\mu^\top)e \\ &= \mathbf{F}(\mathbf{I} - \gamma\mathbf{P}_\pi)e - d_\mu d_\mu^\top e \\ &= (1-\gamma)\mathbf{F}e - d_\mu d_\mu^\top e \\ &= (1-\gamma)f - d_\mu \\ &= (1-\gamma)(\mathbf{I} - \gamma\mathbf{P}_\pi^\top)^{-1}d_\mu - d_\mu \\ &= (1-\gamma)[(\mathbf{I} - \gamma\mathbf{P}_\pi^\top)^{-1} - \mathbf{I}]d_\mu \\ &= (1-\gamma)[\sum_{t=0}^{\infty}(\gamma\mathbf{P}_\pi^\top)^t - \mathbf{I}]d_\mu \\ &= (1-\gamma)[\sum_{t=1}^{\infty}(\gamma\mathbf{P}_\pi^\top)^t]d_\mu > 0, \end{aligned} \tag{15}$$

$$\begin{aligned} &e^\top(\mathbf{F}(\mathbf{I} - \gamma\mathbf{P}_\pi) - \mathbf{d}_\mu d_\mu^\top) \\ &= e^\top\mathbf{F}(\mathbf{I} - \gamma\mathbf{P}_\pi) - e^\top d_\mu d_\mu^\top \\ &= d_\mu^\top - e^\top d_\mu d_\mu^\top \\ &= d_\mu^\top - d_\mu^\top \\ &= 0, \end{aligned} \tag{16}$$

where e is the all-ones vector. (15) and (16) show that the matrix $\mathbf{F}(\mathbf{I} - \gamma\mathbf{P}_\pi) - d_\mu d_\mu^\top$ of diagonal entries are positive and its off-diagonal entries are negative. So its each row sum plus the corresponding column sum is positive. So $\mathbf{A}_{\text{VMETD}}$ is positive definite. □

Optimal Policy Invariance

This section prove the optimal policy invariance of VMTD, VMTDC and VMETD in control experiments, laying the groundwork for subsequent experiments.

As shown in Table 2, although there is a bias between the true value and the predicted value, action a_3 is still chosen under the greedy-policy. On the contrary, supervised learning is usually used to predict temperature, humidity, morbidity, etc. If the bias is too large, the consequences could be serious.

Table 2: Comparison of action selection with and without constant bias in Q values.

ACTION	Q VALUE	Q VALUE WITH BIAS
$Q(s, a_0)$	1	5
$Q(s, a_1)$	2	6
$Q(s, a_2)$	3	7
$Q(s, a_3)$	4	8
$\arg\min_a Q(s, a)$	a_3	a_3

In addition, reward shaping can significantly speed up the learning by adding a shaping reward $F(s, s')$ to the original reward r, where $F(s, s')$ is the general form of any state-based shaping reward. Static potential-based reward shaping (Static PBRS) maintains the policy invariance if the shaping reward follows from $F(s, s') = \gamma f(s') - f(s)$ (Ng, Harada, and Russell 1999).

This means that we can make changes to the TD error $\delta = r + \gamma \boldsymbol{\theta}^\top \boldsymbol{\phi}' - \boldsymbol{\theta}^\top \boldsymbol{\phi}$ while still ensuring the invariance of the optimal policy,

$$\delta - \omega = r + \gamma \boldsymbol{\theta}^\top \boldsymbol{\phi}' - \boldsymbol{\theta}^\top \boldsymbol{\phi} - \omega,$$

where ω is a constant, acting as a static PBRS. This also means that algorithms with the optimization goal of minimizing errors, after introducing reward shaping, may result in larger or smaller bias. Fortunately, as discussed above, bias is acceptable in reinforcement learning. However, the problem is that selecting an appropriate ω requires expert knowledge. This forces us to learn ω dynamically, i.e., $\omega = \omega_t$ and dynamic PBRS can also maintain the policy invariance if the shaping reward is $F(s, t, s', t') = \gamma f(s', t') - f(s, t)$, where t is the time-step the agent reaches in state s (Devlin and Kudenko 2012). However, this result requires the convergence guarantee of the dynamic potential function $f(s, t)$. If $f(s, t)$ does not converge as the time-step $t \to \infty$, the Q-values of dynamic PBRS are not guaranteed to converge.

Let $f_{\omega_t}(s) = \frac{\omega_t}{\gamma - 1}$. Thus, $F_{\omega_t}(s, s') = \gamma f_{\omega_t}(s') - f_{\omega_t}(s) = \omega_t$ is a dynamic PBRS. And if ω converges finally, the dynamic potential function $f(s, t)$ will converge. Bias is the expected difference between the predicted value and the true value. Therefore, under the premise of bootstrapping, we first think of letting $\omega \doteq \mathbb{E}[\delta]$ or $\omega \doteq \mathbb{E}[F\rho\delta]$.

Experimental Studies

This section assesses algorithm performance through experiments, which are divided into policy evaluation experiments and control experiments. The evaluation experimental environments is the 2-state. In a 2-state environment, we conducted two types of experiments—on-policy and off-policy—to verify the relationship between the convergence speed of the algorithm and the smallest eigenvalue of the key matrix $\mathbf{A}$. Control experiments, by allowing the algorithm to interact with the environment to optimize the policy, can evaluate its performance in learning the optimal policy. This provides a more comprehensive assessment of the algorithm's overall capabilities. The control experimental environments are Maze, CliffWalking-v0, MountainCar-v0, and Acrobot-v1. The control algorithms for TDC, ETD, VMTDC, and VMETD are named GQ, EQ, VMGQ, and VMEQ, respectively. For TD and VMTD control algorithms, there are two variants each: Sarsa and Q-learning for TD, and VMSarsa and VMQ for VMTD.

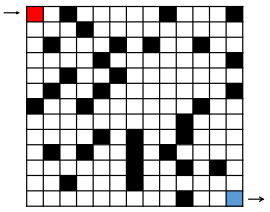


Figure 2: Maze.

Testing Tasks

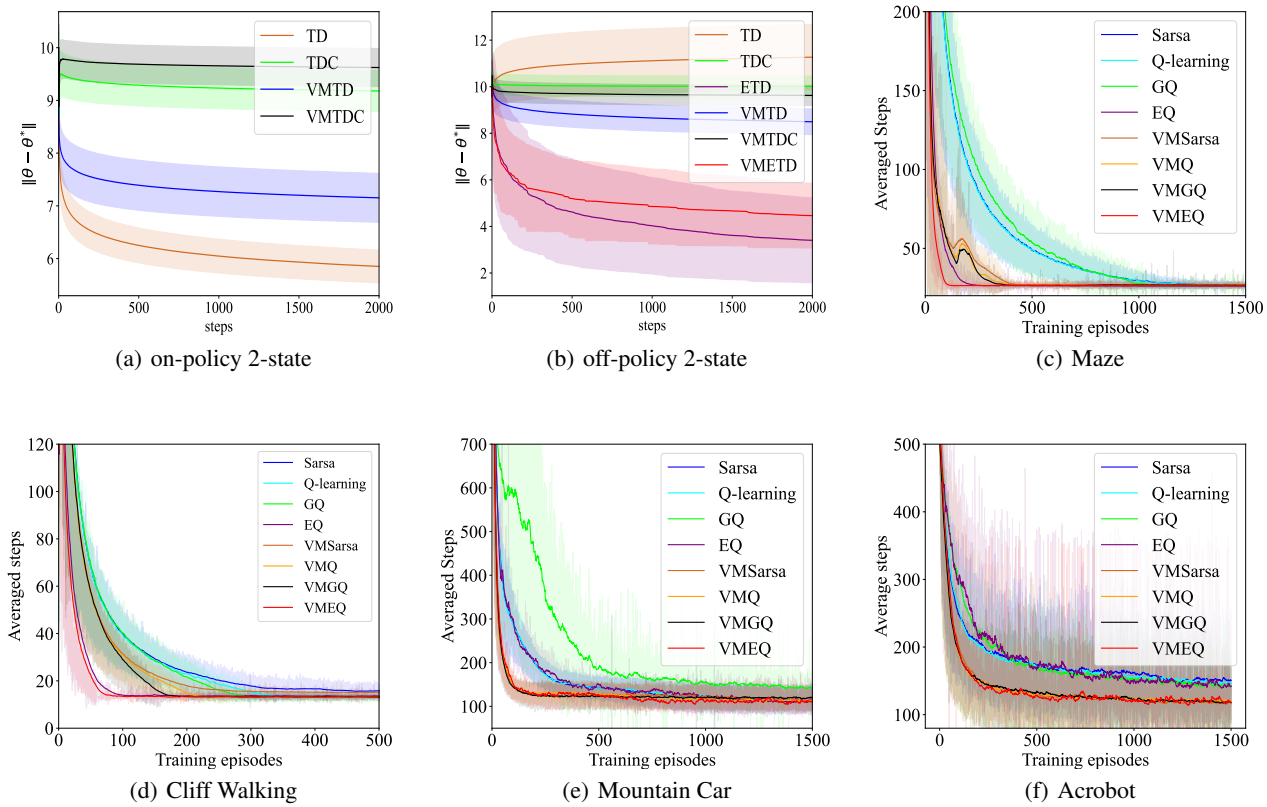
Maze: The learning agent should find a shortest path from the upper left corner to the lower right corner. In each state, there are four alternative actions: up, $down$, $left$, and $right$, which takes the agent deterministically to the corresponding neighbour state, except when a movement is blocked by an obstacle or the edge of the maze. Rewards are -1 in all transitions until the agent reaches the goal state. The discount factor $\gamma = 0.99$, and states s are represented by tabular features.The maximum number of moves in the game is set to 1000.

The other three control environments: Cliff Walking, Mountain Car, and Acrobot are selected from the gym official website and correspond to the following versions: "CliffWalking-v0", "MountainCar-v0" and "Acrobot-v1". For specific details, please refer to the gym official website. The maximum number of steps for the Mountain Car environment is set to 1000, while the default settings are used for the other two environments. In Mountain car and Acrobot, features are generated by tile coding.

For all policy evaluation experiments, each experiment is independently run 100 times. For all control experiments, each experiment is independently run 50 times. For specific experimental parameters, please refer to the appendix.

Experimental Results and Analysis

Figure 3(a) shows the learning curves for the on-policy 2-state policy evaluation experiment. In this setup, the convergence speed of TD, VMTD, TDC, and VMTDC decreases sequentially. Table 1 indicates that the smallest eigenvalue of the key matrix for these four algorithms is greater than



(a) on-policy 2-state (b) off-policy 2-state (c) Maze

(d) Cliff Walking (e) Mountain Car (f) Acrobot

Figure 3: Learning curses of one evaluation environment and four contral environments.

0 and decreases sequentially, which is consistent with the experimental curves and table values.

Figure B displays the learning curves for the off-policy 2-state policy evaluation experiment. In this setup, the convergence speed of ETD, VMETD, VMTD, VMTDC, and TDC decreases sequentially, while TD diverges. Table 1 shows that the smallest eigenvalue of the key matrix for ETD, VMETD, VMTD, VMTDC, and TDC is greater than 0 and decreases sequentially, while the smallest eigenvalue for TD is less than 0. This is consistent with the experimental curves and table values. Remarkably, although VMTD is guaranteed to converge under on-policy conditions, it still converges in the off-policy 2-state scenario. The update formula of VMTD indicates that it is essentially an adjustment and correction of the TD update, with the introduction of the parameter ω making the variance of the gradient estimate more stable, thereby making the update of theta more stable.

Figures 3(c), 3(d), 3(e) and 3(f) show the learning curves for four control experiments. A common feature observed across these experiments is that VMEQ outperforms EQ, VMGQ outperforms GQ, VMQ outperforms Q-learning, and VMSarsa outperforms Sarsa. For the Maze and Cliffwalking experiments, VMEQ demonstrated the best performance with the fastest convergence speed. In the Mountain Car and Acrobot experiments, the performance of the four VM algorithms was nearly identical and all outperformed the other algorithms.

Overall, whether in policy evaluation experiments or control experiments, the VM algorithms have demonstrated superior performance, especially excelling in the control experiments.

Conclusion and Future Work

Value-based reinforcement learning typically aims to minimize error as an optimization objective. As an alternation, this study proposes two new objective functions: VBE and VPBE, and derives an on-policy algorithm: VMTD and two off-policy algorithms: VMTDC and VMETD. All algorithms demonstrated superior performance in policy evaluation and control experiments. Both algorithms demonstrated superior performance in policy evaluation and control experiments. Future work may include, but are not limited to,

- analysis of the convergence rate of VMTDC and VMETD.
- extensions of VBE and VPBE to multi-step returns.
- extensions to nonlinear approximations, such as neural networks.

References

Baird, L.; et al. 1995. Residual algorithms: Reinforcement learning with function approximation. In *Proc. 12th Int. Conf. Mach. Learn.*, 30–37.

Bas-Serrano, J.; Curi, S.; Krause, A.; and Neu, G. 2021. Logistic Q-Learning. In *International Conference on Artificial Intelligence and Statistics*, 3610–3618.

Borkar, V. S. 1997. Stochastic approximation with two time scales. *Syst. & Control Letters*, 29(5): 291–294.

Chen, X.; Ma, X.; Li, Y.; Yang, G.; Yang, S.; and Gao, Y. 2023. Modified Retrace for Off-Policy Temporal Difference Learning. In *Uncertainty in Artificial Intelligence*, 303–312. PMLR.

Devlin, S.; and Kudenko, D. 2012. Dynamic potential-based reward shaping. In *Proc. 11th Int. Conf. Autonomous Agents and Multiagent Systems*, 433–440.

Feng, Y.; Li, L.; and Liu, Q. 2019. A kernel loss for solving the Bellman equation. In *Advances in Neural Information Processing Systems*, 15430–15441.

Givchi, A.; and Palhang, M. 2015. Quasi newton temporal difference learning. In *Asian Conference on Machine Learning*, 159–172.

Hackman, L. 2012. *Faster Gradient-TD Algorithms*. Ph.D. thesis, University of Alberta.

Hallak, A.; Tamar, A.; Munos, R.; and Mannor, S. 2016. Generalized emphatic temporal difference learning: bias-variance analysis. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, 1631–1637.

Johnson, R.; and Zhang, T. 2013. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, 315–323.

Korda, N.; and La, P. 2015. On TD (0) with function approximation: Concentration bounds and a centered variant with exponential convergence. In *International conference on machine learning*, 626–634. PMLR.

Liu, B.; Gemp, I.; Ghavamzadeh, M.; Liu, J.; Mahadevan, S.; and Petrik, M. 2018. Proximal gradient temporal difference learning: Stable reinforcement learning with polynomial sample complexity. *Journal of Artificial Intelligence Research*, 63: 461–494.

Liu, B.; Liu, J.; Ghavamzadeh, M.; Mahadevan, S.; and Petrik, M. 2015. Finite-sample analysis of proximal gradient TD algorithms. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence*, 504–513.

Liu, B.; Liu, J.; Ghavamzadeh, M.; Mahadevan, S.; and Petrik, M. 2016. Proximal Gradient Temporal Difference Learning Algorithms. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 4195–4199.

Ng, A. Y.; Harada, D.; and Russell, S. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proc. 16th Int. Conf. Mach. Learn.*, 278–287.

Pan, Y.; White, A.; and White, M. 2017. Accelerated gradient temporal difference learning. In *Proceedings of the 21st AAAI Conference on Artificial Intelligence*, 2464–2470.

Sutton, R.; Maei, H.; Precup, D.; Bhatnagar, S.; Silver, D.; Szepesvári, C.; and Wiewiora, E. 2009. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proc. 26th Int. Conf. Mach. Learn.*, 993–1000.

Sutton, R. S. 1988. Learning to predict by the methods of temporal differences. *Machine learning*, 3(1): 9–44.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement Learning: An Introduction*. The MIT Press, second edition.

Sutton, R. S.; Maei, H. R.; and Szepesvári, C. 2008. A Convergent $O(n)$ Temporal-difference Algorithm for Off-policy Learning with Linear Function Approximation. In *Advances in Neural Information Processing Systems*, 1609–1616. Cambridge, MA: MIT Press.

Sutton, R. S.; Mahmood, A. R.; and White, M. 2016. An emphatic approach to the problem of off-policy temporal-difference learning. *The Journal of Machine Learning Research*, 17(1): 2603–2631.

Tsitsiklis, J. N.; and Van Roy, B. 1997. Analysis of temporal-diffference learning with function approximation. In *Advances in Neural Information Processing Systems*, 1075–1081.

Xu, T.; Wang, Z.; Zhou, Y.; and Liang, Y. 2019. Reanalysis of Variance Reduced Temporal Difference Learning. In *International Conference on Learning Representations*.

Zhang, S.; and Whiteson, S. 2022. Truncated emphatic temporal difference methods for prediction and control. *The Journal of Machine Learning Research*, 23(1): 6859–6917.