Regularized Off-Policy TD-Learning

Bo Liu, Sridhar Mahadevan
Computer Science Department
University of Massachusetts
Amherst, MA 01003
{boliu, mahadeva}@cs.umass.edu

Ji Liu
Computer Science Department
University of Wisconsin
Madison, WI 53706
ji-liu@cs.wisc.edu

Abstract

We present a novel l_1 regularized off-policy convergent TD-learning method (termed RO-TD), which is able to learn sparse representations of value functions with low computational complexity. The algorithmic framework underlying RO-TD integrates two key ideas: off-policy convergent gradient TD methods, such as TDC, and a convex-concave saddle-point formulation of non-smooth convex optimization, which enables first-order solvers and feature selection using online convex regularization. A detailed theoretical and experimental analysis of RO-TD is presented. A variety of experiments are presented to illustrate the off-policy convergence, sparse feature selection capability and low computational cost of the RO-TD algorithm.

1 Introduction

Temporal-difference (TD) learning is a widely used method in reinforcement learning (RL). Although TD converges when samples are drawn "on-policy" by sampling from the Markov chain underlying a policy in a Markov decision process (MDP), it can be shown to be divergent when samples are drawn "off-policy". Off-policy methods are of wider applications since they are able to learn while executing an exploratory policy, learn from demonstrations, and learn multiple tasks in parallel [2]. Sutton et al. [20] introduced convergent off-policy temporal difference learning algorithms, such as TDC, whose computation time scales linearly with the number of samples and the number of features. Recently, a linear off-policy actor-critic algorithm based on the same framework was proposed in [2].

Regularizing reinforcement learning algorithms leads to more robust methods that can scale up to large problems with many potentially irrelevant features. LARS-TD [7] introduced a popular approach of combining l_1 regularization using Least Angle Regression (LARS) with the least-squares TD (LSTD) framework. Another approach was introduced in [5] (LCP-TD) based on the Linear Complementary Problem (LCP) formulation, an optimization approach between linear programming and quadratic programming. LCP-TD uses "warm-starts", which helps significantly reduce the burden of l_1 regularization. A theoretical analysis of l_1 regularization was given in [4], including error bound analysis with finite samples in the on-policy setting. Another approach integrating the Dantzig Selector with LSTD was proposed in [3], overcoming some of the drawbacks of LARS-TD. An approximate linear programming approach for finding l_1 regularized solutions of the Bellman equation was presented in [17]. All of these approaches are second-order methods, requiring complexity approximately cubic in the number of (active) features. Another approach to feature selection is to greedily add new features, proposed recently in [15]. Regularized first-order reinforcement learning approaches have recently been investigated in the on-policy setting as well, wherein convergence of l_1 regularized temporal difference learning is discussed in [16] and mirror descent [6] is used in [11].

In this paper, the off-policy TD learning problem is formulated from the stochastic optimization perspective. A novel objective function is proposed based on the linear equation formulation of the TDC algorithm. The optimization problem underlying off-policy TD methods, such as TDC, is reformulated as a convex-concave saddle-point stochastic approximation problem, which is both convex and incrementally solvable. A detailed theoretical and experimental study of the RO-TD algorithm is presented.

Here is a brief roadmap to the rest of the paper. Section 2 reviews the basics of MDPs, RL and recent work on off-policy convergent TD methods, such as the TDC algorithm. Section 3 introduces the proximal gradient method and the convex-concave saddle-point formulation of non-smooth convex optimization. Section 4 presents the new RO-TD algorithm. Convergence analysis of RO-TD is presented in Section 5. Finally, in Section 6, experimental results are presented to demonstrate the effectiveness of RO-TD.

2 Reinforcement Learning and the TDC Algorithm

A *Markov Decision Process* (MDP) is defined by the tuple $(S, A, P^a_{ss'}, R, \gamma)$, comprised of a set of states S, a set of (possibly state-dependent) actions A (A_s), a dynamical system model comprised of the transition kernel $P^a_{ss'}$ specifying the probability of transition to state s' from state s under action a, a reward model R, and $0 \leq \gamma < 1$ is a discount factor. A policy $\pi : S \rightarrow A$ is a deterministic mapping from states to actions. Associated with each policy π is a value function V^π, which is the fixed point of the Bellman equation:

$$V^\pi(s) = T^\pi V^\pi(s) = R^\pi(s) + \gamma P^\pi V^\pi(s)$$

where R^π is the expected immediate reward function (treated here as a column vector) and P^π is the state transition function under fixed policy π, and T^π is known as the *Bellman operator*. In what follows, we often drop the dependence of V^π, T^π, R^π on π, for notational simplicity. In linear value function approximation, a value function is assumed to lie in the linear span of a basis function matrix Φ of dimension $|S| \times d$, where d is the number of linear independent features. Hence, $V \approx \hat{V} = \Phi\theta$. The vector space of all value functions is a normed inner product space, where the "length" of any value function f is measured as $||f||^2_\Xi = \sum_s \xi(s) f^2(s) = f'\Xi f$ weighted by Ξ, where Ξ is defined in Figure 1. For the t-th sample, ϕ_t,ϕ'_t, θ_t and δ_t are defined in Figure 1. TD learning uses the following update rule $\theta_{t+1} = \theta_t + \alpha_t \delta_t \phi_t$, where α_t is the stepsize. However, TD is only guaranteed to converge in the on-policy setting, although in many off-policy situations, it still has satisfactory performance [21]. TD with gradient correction (TDC) [20] aims to minimize the mean-square projected Bellman error (MSPBE) in order to guarantee off-policy convergence. MSPBE is defined as

$$\text{MSPBE}(\theta) = \|\Phi\theta - \Pi T(\Phi\theta)\|^2_\Xi = (\Phi^T\Xi(T\Phi\theta - \Phi\theta))^T(\Phi^T\Xi\Phi)^{-1}\Phi^T\Xi(T\Phi\theta - \Phi\theta) \quad (1)$$

To avoid computing the inverse matrix $(\Phi^T\Xi\Phi)^{-1}$ and to avoid the double sampling problem [19] in (1), an auxiliary variable w is defined

$$w \quad = \quad (\Phi^T\Xi\Phi)^{-1}\Phi^T\Xi(T\Phi\theta - \Phi\theta) \quad (2)$$

The two time-scale gradient descent learning method TDC [20] is defined below

$$\theta_{t+1} = \theta_t + \alpha_t\delta_t\phi_t - \alpha_t\gamma\phi_t{}'(\phi_t^T w_t), w_{t+1} = w_t + \beta_t(\delta_t - \phi_t^T w_t)\phi_t \quad (3)$$

where $-\alpha_t\gamma\phi_t{}'(\phi_t^T w_t)$ is the term for correction of gradient descent direction, and $\beta_t = \eta\alpha_t, \eta > 1$.

3 Proximal Gradient and Saddle-Point First-Order Algorithms

We now introduce some background material from convex optimization. The proximal mapping associated with a convex function h is defined as:[1]

$$prox_h(x) = \arg\min_u (h(u) + \frac{1}{2}\|u - x\|^2) \quad (4)$$

[1]The proximal mapping can be shown to be the resolvent of the subdifferential of the function h.

- Ξ is a diagonal matrix whose entries $\xi(s)$ are given by a positive probability distribution over states. $\Pi = \Phi(\Phi^T \Xi \Phi)^{-1} \Phi^T \Xi$ is the weighted least-squares projection operator.
- A square root of A is a matrix B satisfying $B^2 = A$ and B is denoted as $A^{\frac{1}{2}}$. Note that $A^{\frac{1}{2}}$ may not be unique.
- $[\cdot, \cdot]$ is a row vector, and $[\cdot; \cdot]$ is a column vector.
- For the t-th sample, ϕ_t (the t-th row of Φ), ϕ'_t (the t-th row of Φ') are the feature vectors corresponding to s_t, s'_t, respectively. θ_t is the coefficient vector for t-th sample in first-order TD learning methods, and $\delta_t = (r_t + \gamma \phi_t^{'T} \theta_t) - \phi_t^T \theta_t$ is the temporal difference error. Also, $x_t = [w_t; \theta_t]$, α_t is a stepsize, $\beta_t = \eta \alpha_t, \eta > 0$.
- m, n are conjugate numbers if $\frac{1}{m} + \frac{1}{n} = 1, m \geq 1, n \geq 1$. $||x||_m = (\sum_j |x_j|^m)^{\frac{1}{m}}$ is the m-norm of vector x.
- ρ is l_1 regularization parameter, λ is the eligibility trace factor, N is the sample size, d is the number of basis functions, p is the number of active basis functions.

Figure 1: Notation used in this paper.

In the case of $h(x) = \rho \|x\|_1 (\rho > 0)$, which is particularly important for sparse feature selection, the proximal operator turns out to be the soft-thresholding operator $S_\rho(\cdot)$, which is an *entry-wise* shrinkage operator:

$$prox_h(x)_i = S_\rho(x_i) = \max(x_i - \rho, 0) - \max(-x_i - \rho, 0) \tag{5}$$

where i is the index, and ρ is a threshold. With this background, we now introduce the proximal gradient method. If the optimization problem is

$$x^* = \arg\min_{x \in X} (f(x) + h(x)) \tag{6}$$

wherein $f(x)$ is a convex and differentiable loss function and the regularization term $h(x)$ is convex, possibly non-differentiable and computing $prox_h$ is not expensive, then computation of (6) can be carried out using the *proximal gradient method*:

$$x_{t+1} = prox_{\alpha_t h} (x_t - \alpha_t \nabla f(x_t)) \tag{7}$$

where $\alpha_t > 0$ is a (decaying) stepsize, a constant or it can be determined by line search.

3.1 Convex-concave Saddle-Point First Order Algorithms

The key novel contribution of our paper is a convex-concave saddle-point formulation for regularized off-policy TD learning. A convex-concave saddle-point problem is formulated as follows. Let $x \in X, y \in Y$, where X, Y are both nonempty bounded closed convex sets, and $f(x) : X \to \mathbb{R}$ be a convex function. If there exists a function $\varphi(\cdot, \cdot)$ such that $f(x)$ can be represented as $f(x) := \sup_{y \in Y} \varphi(x, y)$, then the pair (φ, Y) is referred as the saddle-point representation of f. The optimization problem of minimizing f over X is converted into an equivalent convex-concave saddle-point problem $SadVal = \inf_{x \in X} \sup_{y \in Y} \varphi(x, y)$ of φ on $X \times Y$. If f is non-smooth yet convex and well structured, which is not suitable for many existing optimization approaches requiring smoothness, its saddle-point representation φ is often smooth and convex. Thus, convex-concave saddle-point problems are, therefore, usually better suited for first-order methods [6]. A comprehensive overview on extending convex minimization to convex-concave saddle-point problems with unified variational inequalities is presented in [1]. As an example, consider $f(x) = ||Ax - b||_m$ which admits a bilinear minimax representation

$$f(x) := \|Ax - b\|_m = \max_{\|y\|_n \leq 1} y^T (Ax - b) \tag{8}$$

where m, n are conjugate numbers. Using the approach in [13], Equation (8) can be solved as

$$x_{t+1} = x_t - \alpha_t A^T y_t, y_{t+1} = \Pi_n(y_t + \alpha_t (A x_t - b)) \tag{9}$$

where Π_n is the projection operator of y onto the unit l_n-ball $\|y\|_n \leq 1$,which is defined as

$$\Pi_n(y) = \min(1, 1/\|y\|_n) y, n = 2, 3, \cdots, \Pi_\infty(y_i) = \min(1, 1/|y_i|) y_i \tag{10}$$

and Π_∞ is an entrywise operator.

4 Regularized Off-policy Convergent TD-Learning

We now describe a novel algorithm, regularized off-policy convergent TD-learning (RO-TD), which combines off-policy convergence and scalability to large feature spaces. The objective function is proposed based on the linear equation formulation of the TDC algorithm. Then the objective function is represented via its dual minimax problem. The RO-TD algorithm is proposed based on the primal-dual subgradient saddle-point algorithm, and inspired by related methods in [12],[13].

4.1 Objective Function of Off-policy TD Learning

In this subsection, we describe the objective function of the regularized off-policy RL problem. We now first formulate the two updates of θ_t, w_t into a single iteration by rearranging the two equations in (3) as $x_{t+1} = x_t - \alpha_t(A_t x_t - b_t)$, where $x_t = [w_t; \theta_t]$,

$$A_t = \begin{bmatrix} \eta\phi_t {\phi_t}^T & \eta\phi_t(\phi_t - \gamma\phi'_t)^T \\ \gamma\phi'_t {\phi_t}^T & \phi_t(\phi_t - \gamma\phi'_t)^T \end{bmatrix}, b_t = \begin{bmatrix} \eta r_t \phi_t \\ r_t \phi_t \end{bmatrix} \tag{11}$$

Following [20], the TDC algorithm solution follows from the linear equation $Ax = b$, where

$$A = \mathbb{E}[A_t], b = \mathbb{E}[b_t], x = [w; \theta] \tag{12}$$

There are some issues regarding the objective function, which arise from the online convex optimization and reinforcement learning perspectives, respectively. The first concern is that the objective function should be convex and stochastically solvable. Note that A, A_t are neither PSD nor symmetric, and it is not straightforward to formulate a convex objective function based on them. The second concern is that since we do not have knowledge of A, the objective function should be separable so that it is stochastically solvable based on A_t, b_t. The other concern regards the sampling condition in temporal difference learning: double-sampling. As pointed out in [19], double-sampling is a necessary condition to obtain an unbiased estimator if the objective function is the Bellman residual or its derivatives (such as projected Bellman residual), wherein the product of Bellman error or projected Bellman error metrics are involved. To overcome this sampling condition constraint, the product of TD errors should be avoided in the computation of gradients. Consequently, based on the linear equation formulation in (12) and the requirement on the objective function discussed above, we propose the regularized loss function as

$$L(x) = \|Ax - b\|_m + h(x) \tag{13}$$

Here we also enumerate some intuitive objective functions and give a brief analysis on the reasons why they are not suitable for regularized off-policy first-order TD learning. One intuitive idea is to add a sparsity penalty on MSPBE, i.e., $L(\theta) = \text{MSPBE}(\theta) + \rho\|\theta\|_1$. Because of the l_1 penalty term, the solution to $\nabla L = 0$ does not have an analytical form and is thus difficult to compute. The second intuition is to use the online least squares formulation of the linear equation $Ax = b$. However, since A is not symmetric and positive semi-definite (PSD), $A^{\frac{1}{2}}$ does not exist and thus $Ax = b$ cannot be reformulated as $\min_{x \in X} ||A^{\frac{1}{2}}x - A^{-\frac{1}{2}}b||_2^2$. Another possible idea is to attempt to find an objective function whose gradient is exactly $A_t x_t - b_t$ and thus the regularized gradient is $prox_{\alpha_t h(x_t)}(A_t x_t - b_t)$. However, since A_t is not symmetric, this gradient does not explicitly correspond to any kind of optimization problem, not to mention a convex one[2].

4.2 RO-TD Algorithm Design

In this section, the problem of (13) is formulated as a convex-concave saddle-point problem, and the RO-TD algorithm is proposed. Analogous to (8), the regularized loss function can be formulated as

$$\|Ax - b\|_m + h(x) = \max_{\|y\|_n \leq 1} y^T(Ax - b) + h(x) \tag{14}$$

Similar to (9), Equation (14) can be solved via an iteration procedure as follows, where $x_t = [w_t; \theta_t]$.

$$\begin{aligned} x_{t+\frac{1}{2}} &= x_t - \alpha_t A_t^T y_t \quad , \quad y_{t+\frac{1}{2}} = y_t + \alpha_t(A_t x_t - b_t) \\ x_{t+1} &= prox_{\alpha_t h}(x_{t+\frac{1}{2}}) \quad , \quad y_{t+1} = \Pi_n(y_{t+\frac{1}{2}}) \end{aligned} \tag{15}$$

[2]Note that the A matrix in GTD2's linear equation representation is symmetric, yet is not PSD, so it cannot be formulated as a convex problem.

The averaging step, which plays a crucial role in stochastic optimization convergence, generates the *approximate saddle-points* [6, 12]

$$\bar{x}_t = \left(\sum_{i=0}^{t} \alpha_i\right)^{-1} \sum_{i=0}^{t} \alpha_i x_i, \bar{y}_t = \left(\sum_{i=0}^{t} \alpha_i\right)^{-1} \sum_{i=0}^{t} \alpha_i y_i \tag{16}$$

Due to the computation of A_t in (15) at each iteration, the computation cost appears to be $O(Nd^2)$, where N, d are defined in Figure 1. However, the computation cost is actually $O(Nd)$ with a linear algebraic trick by computing not A_t but $y_t^T A_t, A_t x_t - b_t$. Denoting $y_t = [y_{1,t}; y_{2,t}]$, where $y_{1,t}; y_{2,t}$ are column vectors of equal length, we have

$$y_t^T A_t = \left[\begin{array}{cc} \eta \phi_t^T (y_{1,t}^T \phi_t) + \gamma \phi_t^T (y_{2,t}^T \phi_t') & (\phi_t - \gamma \phi_t')^T (\eta y_{1,t}^T + y_{2,t}^T) \phi_t \end{array} \right] \tag{17}$$

$A_t x_t - b_t$ can be computed according to Equation (3) as follows:

$$A_t x_t - b_t = \left[\begin{array}{c} -\eta(\delta_t - \phi_t^T w_t)\phi_t; \gamma(\phi_t^T w_t)\phi_t{}' - \delta_t \phi_t \end{array} \right] \tag{18}$$

Both (17) and (18) are of linear computation complexity. Now we are ready to present the RO-TD algorithm:

Algorithm 1 RO-TD

Let π be some fixed policy of an MDP M, and let the sample set $S = \{s_i, r_i, s_i'\}_{i=1}^N$. Let Φ be some fixed basis.

1: **repeat**
2: Compute $\phi_t, \phi_t{}'$ and TD error $\delta_t = (r_t + \gamma \phi_t'^T \theta_t) - \phi_t^T \theta_t$
3: Compute $y_t^T A_t, A_t x_t - b_t$ in Equation (17) and (18).
4: Compute x_{t+1}, y_{t+1} as in Equation (15)
5: Set $t \leftarrow t + 1$;
6: **until** $t = N$;
7: Compute $\bar{x}_N, \bar{y}_N$ as in Equation (16) with $t = N$

There are some design details of the algorithm to be elaborated. First, the regularization term $h(x)$ can be any kind of convex regularization, such as ridge regression or sparsity penalty $\rho||x||_1$. In case of $h(x) = \rho||x||_1$, $prox_{\alpha_t h}(\cdot) = S_{\alpha_t \rho}(\cdot)$. In real applications the sparsification requirement on θ and auxiliary variable w may be different, i.e., $h(x) = \rho_1 \|\theta\|_1 + \rho_2 \|w\|_1, \rho_1 \neq \rho_2$, one can simply replace the uniform soft thresholding $S_{\alpha_t \rho}$ by two separate soft thresholding operations $S_{\alpha_t \rho_1}, S_{\alpha_t \rho_2}$ and thus the third equation in (15) is replaced by the following,

$$x_{t+\frac{1}{2}} = \left[w_{t+\frac{1}{2}}; \theta_{t+\frac{1}{2}}\right], \theta_{t+1} = S_{\alpha_t \rho_1}(\theta_{t+\frac{1}{2}}), w_{t+1} = S_{\alpha_t \rho_2}(w_{t+\frac{1}{2}}) \tag{19}$$

Another concern is the choice of conjugate numbers (m, n). For ease of computing Π_n, we use $(2, 2)(l_2$ fit), $(+\infty, 1)$(uniform fit) or $(1, +\infty)$. $m = n = 2$ is used in the experiments below.

4.3 RO-GQ(λ) Design

GQ(λ)[10] is a generalization of the TDC algorithm with eligibility traces and off-policy learning of temporally abstract predictions, where the gradient update changes from Equation (3) to

$$\theta_{t+1} = \theta_t + \alpha_t[\delta_t e_t - \gamma(1-\lambda) w_t{}^T e_t \bar{\phi}_{t+1}], w_{t+1} = w_t + \beta_t(\delta_t e_t - w_t^T \phi_t \phi_t) \tag{20}$$

The central element is to extend the MSPBE function to the case where it incorporates eligibility traces. The objective function and corresponding linear equation component A_t, b_t can be written as follows:

$$L(\theta) = ||\Phi\theta - \Pi T^{\pi\lambda} \Phi\theta||_\Xi^2 \tag{21}$$

$$A_t = \left[\begin{array}{cc} \eta \phi_t \phi_t{}^T & \eta e_t (\phi_t - \gamma \bar{\phi}_{t+1})^T \\ \gamma(1-\lambda)\bar{\phi}_{t+1} e_t^T & e_t(\phi_t - \gamma \bar{\phi}_{t+1})^T \end{array} \right], b_t = \left[\begin{array}{c} \eta r_t e_t \\ r_t e_t \end{array} \right] \tag{22}$$

Similar to Equation (17) and (18), the computation of $y_t^T A_t, A_t x_t - b_t$ is

$$\begin{aligned} y_t^T A_t &= \left[\eta \phi_t^T (y_{1,t}^T \phi_t) + \gamma(1-\lambda) e_t^T (y_{2,t}^T \bar{\phi}_{t+1}) \quad (\phi_t - \gamma \bar{\phi}_{t+1})^T (\eta y_{1,t}^T + y_{2,t}^T) e_t \right] \\ A_t x_t - b_t &= \left[-\eta(\delta_t e_t - \phi_t^T w_t \phi_t); \gamma(1-\lambda)(e_t^T w_t)\bar{\phi}_{t+1} - \delta_t e_t \right] \end{aligned} \tag{23}$$

where eligibility traces e_t, and $\bar{\phi}_t, T^{\pi\lambda}$ are defined in [10]. Algorithm 2, RO-GQ(λ), extends the RO-TD algorithm to include eligibility traces.

Algorithm 2 RO-GQ(λ)

Let π and Φ be as defined in Algorithm 1. Starting from s_0.

1: **repeat**
2: Compute $\phi_t, \bar{\phi}_{t+1}$ and TD error $\delta_t = (r_t + \gamma \bar{\phi}_{t+1}^T \theta_t) - \phi_t^T \theta_t$
3: Compute $y_t^T A_t, A_t x_t - b_t$ in Equation (23).
4: Compute x_{t+1}, y_{t+1} as in Equation (15)
5: Choose action a_t, and get s_{t+1}
6: Set $t \leftarrow t + 1$;
7: **until** s_t is an absorbing state;
8: Compute $\bar{x}_t, \bar{y}_t$ as in Equation (16)

4.4 Extension

It is also worth noting that there exists another formulation of the loss function different from Equation (13) with the following convex-concave formulation as in [14, 6],

$$\begin{aligned} \min_x \frac{1}{2} \|Ax - b\|_2^2 + \rho \|x\|_1 &= \max_{\|A^T y\|_\infty \leq 1} (b^T y - \frac{\rho}{2} y^T y) \\ &= \min_x \max_{\|u\|_\infty \leq 1, y} \Big(x^T u + y^T (Ax - b) - \frac{\rho}{2} y^T y \Big) \end{aligned} \tag{24}$$

which can be solved iteratively without the proximal gradient step as follows, which serves as a counterpart of Equation (15),

$$\begin{aligned} x_{t+1} = x_t - \alpha_t \rho (u_t + {A_t}^T y_t) \quad , \quad y_{t+1} = y_t + \frac{\alpha_t}{\rho} (A_t x_t - b_t - \rho y_t) \\ u_{t+\frac{1}{2}} = u_t + \frac{\alpha_t}{\rho} x_t \quad , \quad u_{t+1} = \Pi_\infty (u_{t+\frac{1}{2}}) \end{aligned} \tag{25}$$

5 Convergence Analysis of RO-TD

Assumption 1 (**MDP**)[20]: The underlying Markov Reward Process (MRP) $M = (S, P, R, \gamma)$ is finite and mixing, with stationary distribution π. Assume that $\exists$ a scalar $R_{\max}$ such that $Var[r_t | s_t] \leq R_{\max}$ holds w.p.1.

Assumption 2 (**Basis Function**)[20]: Φ is a full column rank matrix, namely, Φ comprises a linear independent set of basis functions w.r.t all sample states in sample set S. Also, assume the features $(\phi_t, \phi_t^{'})$ have uniformly bounded second moments. Finally, if $(s_t, a_t, s_t^{'})$ is an i.i.d sequence, $\forall t, \|\phi_t\|_\infty < +\infty, \|\phi_t'\|_\infty < +\infty$.

Assumption 3 (**Subgradient Boundedness**)[12]: Assume for the bilinear convex-concave loss function defined in (14), the sets X, Y are closed compact sets. Then the subgradient $y_t^T A_t$ and $A_t x_t - b_t$ in RO-TD algorithm are uniformly bounded, i.e., there exists a constant L such that $\|A_t x_t - b_t\| \leq L, \|y_t^T A_t\| \leq L$.

Proposition 1: The approximate saddle-point $\bar{x}_t$ of RO-TD converges w.p.1 to the global minimizer of the following,

$$x^* = \arg\min_{x \in X} \|Ax - b\|_m + \rho \|x\|_1 \tag{26}$$

Proof Sketch: See the supplementary material for details.

6 Empirical Results

We now demonstrate the effectiveness of the RO-TD algorithm against other algorithms across a number of benchmark domains. LARS-TD [7], which is a popular second-order sparse reinforcement learning algorithm, is used as the baseline algorithm for feature selection and TDC is used as the off-policy convergent RL baseline algorithm, respectively.

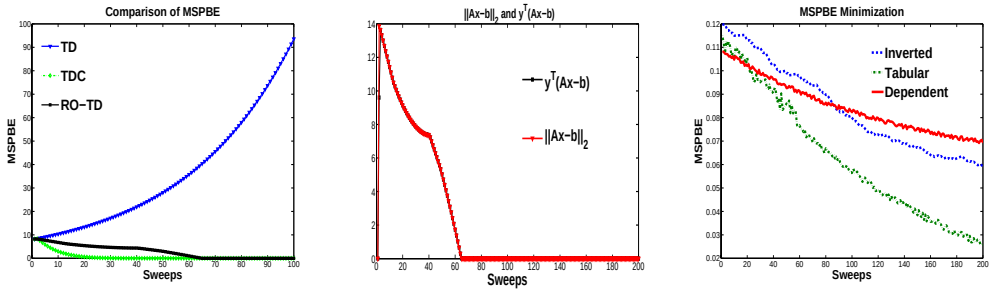

Figure 2: Illustrative examples of the convergence of RO-TD using the Star and Random-walk MDPs.

6.1 MSPBE Minimization and Off-Policy Convergence

This experiment aims to show the minimization of MSPBE and off-policy convergence of the RO-TD algorithm. The 7 state star MDP is a well known counterexample where TD diverges monotonically and TDC converges. It consists of 7 states and the reward w.r.t any transition is zero. Because of this, the star MDP is unsuitable for LSTD-based algorithms, including LARS-TD since $\Phi^T R = 0$ always holds. The random-walk problem is a standard Markov chain with 5 states and two absorbing state at two ends. Three sets of different bases Φ are used in [20], which are tabular features, inverted features and dependent features respectively. An identical experiment setting to [20] is used for these two domains. The regularization term $h(x)$ is set to 0 to make a fair comparison with TD and TDC. $\alpha = 0.01$, $\eta = 10$ for TD, TDC and RO-TD. The comparison with TD, TDC and RO-TD is shown in the left subfigure of Figure 2, where TDC and RO-TD have almost identical MSPBE over iterations. The middle subfigure shows the value of $y_t^T(Ax_t - b)$ and $\|Ax_t - b\|_2$, wherein $\|Ax_t - b\|_2$ is always greater than the value of $y_t^T(Ax_t - b)$. Note that for this problem, the Slater condition is satisfied so there is no duality gap between the two curves. As the result shows, TDC and RO-TD perform equally well, which illustrates the off-policy convergence of the RO-TD algorithm. The result of random-walk chain is averaged over 50 runs. The rightmost subfigure of Figure 2 shows that RO-TD is able to reduce MSPBE over successive iterations w.r.t three different basis functions.

6.2 Feature Selection

In this section, we use the mountain car example with a variety of bases to show the feature selection capability of RO-TD. The Mountain car MDPis an optimal control problem with a continuous two-dimensional state space. The steep discontinuity in the value function makes learning difficult for bases with global support. To make a fair comparison, we use the same basis function setting as in [7], where two dimensional grids of $2, 4, 8, 16, 32$ RBFs are used so that there are totally 1365 basis functions. For LARS-TD, 500 samples are used. For RO-TD and TDC, 3000 samples are used by executing 15 episodes with 200 steps for each episode, stepsize $\alpha_t = 0.001$, and $\rho_1 = 0.01, \rho_2 = 0.2$. We use the result of LARS-TD and l_2 LSTD reported in [7]. As the result shows in Table 1, RO-TD is able to perform feature selection successfully, whereas TDC and TD failed. It is worth noting that comparing the performance of RO-TD and LARS-TD is not the focus of this paper since LARS-TD is not convergent off-policy and RO-TD's performance can be further optimized using the mirror-descent approach with the Mirror-Prox algorithm [6] which incorporates mirror descent with an extragradient [9], as discussed below.

Algorithm	LARS-TD	RO-TD	l_2 LSTD	TDC	TD
Success(20/20)	100%	100%	0%	0%	0%
Steps	142.25 ± 9.74	147.40 ± 13.31	-	-	-

Table 1: Comparison of TD, LARS-TD, RO-TD, l_2 LSTD, TDC and TD

Experiment\Method	RO-GQ(λ)	GQ(λ)	LARS-TD
Experiment 1	6.9 ± 4.82	11.3 ± 9.58	-
Experiment 2	14.7 ± 10.70	27.2 ± 6.52	-

Table 2: Comparison of RO-GQ(λ), GQ(λ), and LARS-TD on Triple-Link Inverted Pendulum Task showing minimum number of learning episodes.

6.3 High-dimensional Under-actuated Systems

The triple-link inverted pendulum [18] is a highly nonlinear under-actuated system with 8-dimensional state space and discrete action space. The state space consists of the angles and angular velocity of each arm as well as the position and velocity of the car. The discrete action space is $\{0, 5\text{Newton}, -5\text{Newton}\}$. The goal is to learn a policy that can balance the arms for N_x steps within some minimum number of learning episodes. The allowed maximum number of episodes is 300. The pendulum initiates from zero equilibrium state and the first action is randomly chosen to push the pendulum away from initial state. We test the performance of RO-GQ(λ), GQ(λ) and LARS-TD. Two experiments are conducted with $N_x = 10,000$ and $100,000$, respectively. Fourier basis [8] with order 2 is used, resulting in 6561 basis functions. Table 2 shows the results of this experiment, where RO-GQ(λ) performs better than other approaches, especially in Experiment 2, which is a harder task. LARS-TD failed in this domain, which is mainly not due to LARS-TD itself but the quality of samples collected via random walk.

To sum up, RO-GQ(λ) tends to outperform GQ(λ) in all aspects, and is able to outperform LARS-TD based policy iteration in high dimensional domains, as well as in selected smaller MDPs where LARS-TD diverges (e.g., the star MDP). It is worth noting that the computation cost of LARS-TD is $O(Ndp^3)$, where that for RO-TD is $O(Nd)$. If p is linear or sublinear w.r.t d, RO-TD has a significant advantage over LARS-TD. However, compared with LARS-TD, RO-TD requires fine tuning the parameters of α_t, ρ_1, ρ_2 and is usually not as sample efficient as LARS-TD. We also find that tuning the sparsity parameter ρ_2 generates an interpolation between GQ(λ) and TD learning, where a large ρ_2 helps eliminate the correction term of TDC update and make the update direction more similar to the TD update.

7 Conclusions

This paper presents a novel unified framework for designing regularized off-policy convergent RL algorithms combining a convex-concave saddle-point problem formulation for RL with stochastic first-order methods. A detailed experimental analysis reveals that the proposed RO-TD algorithm is both off-policy convergent and is robust to noisy features. There are many interesting future directions for this research. One direction for future work is to extend the subgradient saddle-point solver to a more generalized mirror descent framework. Mirror descent is a generalization of subgradient descent with non-Euclidean distance [1], and has many advantages over gradient descent in high-dimensional spaces. In [6], two algorithms to solve the bilinear saddle-point formulation are proposed based on mirror descent and the extragradient [9], such as the Mirror-Prox algorithm. [6] also points out that the Mirror-Prox algorithm may be further optimized via randomization. To scale to larger MDPs, it is possible to design SMDP-based mirror-descent methods as well.

Acknowledgments

This material is based upon work supported by the Air Force Office of Scientific Research (AFOSR) under grant FA9550-10-1-0383, and the National Science Foundation under Grant Nos. NSF CCF-1025120, IIS-0534999, IIS-0803288, and IIS-1216467 Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the AFOSR or the NSF. We thank M. F. Duarte for helpful discussions.

References

[1] A. Ben-Tal and A. Nemirovski. Non-Euclidean restricted memory level method for large-scale convex optimization. *Mathematical Programming*, 102(3):407–456, 2005.

[2] T. Degris, M. White, and R. S. Sutton. Linear off-policy actor-critic. In *International Conference on Machine Learning*, 2012.

[3] M. Geist, B. Scherrer, A. Lazaric, and M. Ghavamzadeh. A Dantzig Selector Approach to Temporal Difference Learning. In *International Conference on Machine Learning*, 2012.

[4] M. Ghavamzadeh, A. Lazaric, R. Munos, and M. Hoffman. Finite-Sample Analysis of Lasso-TD . In *Proceedings of the 28th International Conference on Machine Learning*, 2011.

[5] J. Johns, C. Painter-Wakefield, and R. Parr. Linear complementarity for regularized policy evaluation and improvement. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2010.

[6] A. Juditsky and A. Nemirovski. *Optimization for Machine Learning*, chapter First-Order Methods for Nonsmooth Convex Large-Scale Optimization. MIT Press, 2011.

[7] J. Zico Kolter and A. Y. Ng. Regularization and feature selection in least-squares temporal difference learning. In *Proceedings of 27 th International Conference on Machine Learning*, 2009.

[8] G. Konidaris, S. Osentoski, and PS Thomas. Value function approximation in reinforcement learning using the fourier basis. In *Proceedings of the Twenty-Fifth Conference on Artificial Intelligence*, 2011.

[9] G. M. Korpelevich. The extragradient method for finding saddle points and other problems. 1976.

[10] H.R. Maei and R.S. Sutton. GQ (λ): A general gradient algorithm for temporal-difference prediction learning with eligibility traces. In *Proceedings of the Third Conference on Artificial General Intelligence*, pages 91–96, 2010.

[11] S. Mahadevan and B. Liu. Sparse Q-learning with Mirror Descent. In *Proceedings of the Conference on Uncertainty in AI*, 2012.

[12] A. Nedić and A. Ozdaglar. Subgradient methods for saddle-point problems. *Journal of optimization theory and applications*, 142(1):205–228, 2009.

[13] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19:1574–1609, 2009.

[14] Y. Nesterov. Gradient methods for minimizing composite objective function. In *www.optimization-online.org*, 2007.

[15] C. Painter-Wakefield and R. Parr. Greedy algorithms for sparse reinforcement learning. In *International Conference on Machine Learning*, 2012.

[16] C. Painter-Wakefield and R. Parr. L1 regularized linear temporal difference learning. Technical report, Duke CS Technical Report TR-2012-01, 2012.

[17] M. Petrik, G. Taylor, R. Parr, and S. Zilberstein. Feature selection using regularization in approximate linear programs for Markov decision processes. In *Proceedings of the International Conference on Machine learning (ICML)*, 2010.

[18] J. Si and Y. Wang. Online learning control by association and reinforcement. *IEEE Transactions on Neural Networks*, 12:264–276, 2001.

[19] R. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[20] R.S. Sutton, H.R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, and E. Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *International Conference on Machine Learning*, pages 993–1000, 2009.

[21] J. Zico Kolter. The Fixed Points of Off-Policy TD. In *Advances in Neural Information Processing Systems 24*, pages 2169–2177, 2011.