Fast Gradient-Descent Methods for Temporal-Difference Learning with Linear Function Approximation

SUTTON, MAEI, PRECUP, SZEPESVARI, ...

Keywords: reinforcement learning, approximate dynamic programming, temporal-difference learning, gradient descent, stochastic approximation, off-policy learning, linear function approximation

Abstract

Sutton, Szepesvári and Maei (2009) recently introduced the first temporal-difference learning algorithm compatible with both linear function approximation and off-policy training, and whose complexity scales only linearly in the size of the function approximator. Although their "gradient temporal difference" (GTD) algorithm converges reliably, it can be very slow compared to conventional linear TD (on on-policy problems where TD is convergent), calling into question its practical utility. In this paper we introduce two new algorithms. The first algorithm, *GTD-2*, is derived and proved convergent just as GTD was, but uses a different objective function and converges significantly faster (but still not as fast as conventional TD). The second new algorithm, *linear TD with gradient correction*, uses the same update rule as conventional TD except for an additional term, which is initially zero. In our experiments, this algorithm converges just as fast as conventional TD in on-policy problems. This algorithm seems to extend linear TD to off-policy learning with no penalty in performance while only doubling computational requirements.

1. Motivation

Temporal-difference methods based on gradient descent and linear function approximation form a core part of the modern field of reinforcement learning and are essential to many of its large-scale applications. However, the simplest and most popular methods, including TD(λ), Q-learning, and Sarsa, are not true gradient-descent methods (Barnard 1993) and, as a result, the conditions under which they converge are narrower and less robust than can usually be guaranteed for gradient-descent methods. In particular, convergence cannot be guaranteed for these methods when they are used with off-policy training (Baird 1995). Off-policy training—training on data from one policy in order to learn the value of another—is useful in dealing with the exploration-exploitation tradeoff and for intra-option learning (Sutton, Precup & Singh 1999). Finding an off-policy temporal-difference algorithm that is effective on large applications with linear function approximation has been one of the most important open problems in reinforcement learning for more than a decade.

Several non-gradient-descent approaches to this problem have been developed, but none has been completely satisfactory. Second-order methods, such as LSTD (Bradtke & Barto 1996; Boyan 1999), can be guaranteed stable under general conditions, but their computational complexity is $O(n^2)$, where n is the number of features used in the linear approximator, as compared to $O(n)$ for TD(λ) and the other simple methods. In applications, n is often too large (e.g., $n = 10^6$ in Computer Go, Silver et al. 2007) for second-order methods to be feasible. Incremental methods such as iLSTD (Geramifard et al. 2006) reduce per-time-step computation to $O(n)$, but still require $O(n^2)$ memory. Precup and colleagues (2001; 2006) have explored $O(n)$ solutions based on importance sampling, but these methods still have the potential for instability, converge slowly, or apply only to special cases.

In this paper we explore the development of true stochastic gradient-descent algorithms for temporal-difference learning with linear function approximation, building on the work of Baird (1995; 1999) and of Sutton, Szepesvári and Maei (2009).

2. Linear value-function approximation

We consider a prototypical case of temporal-difference learning, that of learning a linear approximation to the state-value function for a given policy and Markov decision process (MDP) from sample transitions. We take both the MDP and the policy to be stationary, so their combination determines the stochastic dynamics of a Markov chain. The

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

state of the chain at each time is a random variable, denoted $s_t \in \{1, 2, ..., N\}$, and the state-transition probabilities are given by a matrix P. On each transition, from s_t to s_{t+1}, there is also a reward r_{t+1}, whose distribution depends on both states. We seek to learn the parameter $\theta \in \Re^n$ of an approximate value function $V_\theta : \mathcal{S} \to \Re$ such that

$$V_\theta(s) = \theta^\top \phi_s \approx V(s) = E\left\{\sum_{t=0}^{\infty} \gamma^t r_{t+1} \mid s_0 = s\right\}, \quad (1)$$

where $\phi_s \in \Re^n$ is a feature vector characterizing state s, and $\gamma \in [0, 1)$ is a constant called the *discount rate*.

In this paper we consider *one-step* temporal-difference learning (corresponding to $\lambda = 0$ in TD(λ)), in which there is one independent update to θ for each state transition and associated reward. There are several settings corresponding to how the state transitions are generated. In the on-policy setting, for example, the state transitions come directly from the continuing evolution of the Markov chain. We assume that the Markov chain is ergodic and uni-chain, so there exists a limiting distribution d such that $d_s = \lim_{t\to\infty} \mathbb{P}(s_t = s)$.[1] In the on-policy case, d is linked to the transition probabilities (in particular, we know that $P^\top d = d$) and this linkage is critical to the convergence of algorithms such as conventional TD(λ). In this paper, we consider a general i.i.d. setting, in which the first state of each transition is chosen i.i.d. according to an arbitrary distribution d that may be unrelated to P (this corresponds to off-policy learning). This setting defines a probability over independent triples of state, next state, and reward random variables, denoted (s_k, s'_k, r_k), with associated feature-vector random variables $\phi_k = \phi_{s_k}$ and $\phi'_k = \phi_{s'_k}$. From these we can define, for example, the temporal-difference error,

$$\delta_k = r_k + \gamma\theta_k^\top \phi'_k - \theta_k^\top \phi_k,$$

used in the conventional linear TD(0) algorithm (Sutton, 1988):

$$\theta_{k+1} \leftarrow \theta_k + \alpha_k \delta_k \phi_k, \quad (2)$$

where α_k is a sequence of positive step-sizes.

3. Objective functions

The objective function is whatever function of the modifiable parameter θ we seek to minimize by updating θ. In gradient descent, the updates to θ are then proportional to the gradient or sample gradient of the objective function with respect to θ. The first question then, is what to use for the objective function? For example, one natural choice might be the mean squared error (MSE) between the approximate value function V_θ and the true value function V, averaged over the state space according to how often each state occurs. The MSE objective function is

$$\begin{aligned} \text{MSE}(\theta) &= \sum_s d_s \left(V_\theta(s) - V(s)\right)^2 \\ &= \| V_\theta - V \|_D^2 \end{aligned}$$

In the second equation, V_θ and V are viewed as vectors with one element for each state, and the norm $\| v \|_D^2 = v^\top D v$ is weighted by the matrix D that has the d_s on its diagonal.

In temporal-difference methods, the idea is instead to use an objective function representing how closely the approximate value function satisfies the Bellman equation. The true value function V satisfies the Bellman equation exactly:

$$\begin{aligned} V &= R + \gamma P V \\ &= TV \end{aligned}$$

where R is the vector with components $E\{r_{t+1} \mid s_t = s\}$ and T is known as the *Bellman operator*. A seemingly natural measure of how closely the approximation V_θ satisfies the Bellman equation is the *mean-square Bellman error*:

$$\text{MSBE}(\theta) = \| V_\theta - TV_\theta \|_D^2 . \quad (3)$$

This is the objective function used by the most important prior effort to develop gradient-descent algorithms, that by Baird (1995, 1999). However, most temporal-difference algorithms, including TD(0), LSTD, and GTD(0) do not converge to the minimum of the MSBE. To understand this, note that the Bellman operator follows the underlying state dynamics of the Markov chain, irrespective of the structure of the function approximator. As a result, TV_θ will typically not be representable as V_θ for any θ. Consider the projection operator Π which takes any value function v and projects it to the nearest value function representable by the function approximator:

$$\Pi v = V_\theta \text{ where } \theta = \arg\min_\theta \| V_\theta - v \|_D^2 .$$

In a linear architecture, in which $V_\theta = \Phi\theta$ (where Φ is the matrix whose rows are the ϕ_s), the projection operator is linear and independent of θ:

$$\Pi = \Phi(\Phi^\top D \Phi)^{-1} \Phi^\top D$$

All the algorithms mentioned above find or converge to a fixed point of the composed projection and Bellman operators, that is to a value of θ such that

$$V_\theta = \Pi T V_\theta. \quad (4)$$

We call this value of θ the TD fixed point. In the current work, we take as our objective the deviation from this fixed

[1] Our results apply also to the episodic case if d_s is taken to be the fraction of time steps in state s. In this case, the sum in (1) is only over a finite number of time steps, the columns of P may sum to less than 1, and γ may be equal to 1 (as long as $(\gamma P)^\infty = 0$).

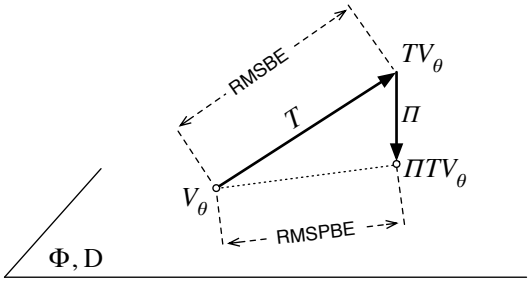

Figure 1. Geometric relationships between (the square root of) the two Bellman-error objective functions.

point. That is, we use as our objective function the mean-square *projected* Bellman error:

$$\text{MSPBE}(\theta) \quad = \quad \| V_\theta - \Pi T V_\theta \|^2_D \,. \tag{5}$$

Figure 1 shows the relationship between this and the MSBE objective function geometrically.

Further insight can be gained by considering the episodic examples in Figure 2. In the system on the left, trajectories start in state A and then either terminate immediately with a reward of zero, or transition to state B with a reward of zero and then terminate with a reward of 1. The two choices occur each with 50% probability, and $\gamma = 1$, so the right values for states A and B are clearly 0.5 and 1 respectively (these values minimize both MSBE and MSPBE). Dayan (1992) used this example to show that a naive gradient-descent approach (based on gradient descent in the mean-squared TD error, $\mathbb{E}\big[\delta^2\big]$) works poorly in that it ends up assigning values of $1/3$ and $2/3$ to A and B even in the tabular case. The example also illustrates the need for two independent samples in the residual-gradient algorithm (Baird 1995) as, with a single example, that algorithm finds the $1/3, 2/3$ solution. With two samples, residual gradient correctly finds the $0.5, 1$ solution. However, consider now the example in the right panel. Here function approximation is in play, in that we have two states, A1 and A2, that share the same feature representation; they look the same and must be given the same approximate value. Trajectories start in each of the two A states with 50% probability; one leads deterministically to B and 1, while the other leads deterministically to 0. From the observed feature vectors, this example looks like the previous, except that here taking multiple samples is no help as the system is deterministic and they will all be the same. Because of this, the residual-gradient algorithm will find the $1/3, 2/3$ solution here. However, the problem is not with the algorithm, but with the objective. The $1/3, 2/3$ solution is in fact the minimum-MSBE solution on this problem; only the MSPBE criterion puts the minimum at $0.5, 1$ on this problem. The MSBE objective causes function approximation resources to be expended trying to reduce the Bellman error associated with

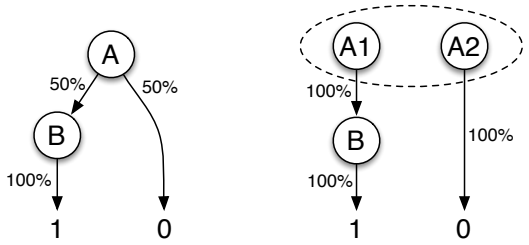

Figure 2. The A-split (left) and split-A (right) examples.

A1 and A2, whereas the MSPBE objective takes into account that their approximated values will ultimately be projected onto the same point.

Finally, we close this discussion of objective functions by giving the function used to derive the original GTD algorithm. This objective function does not seem to have a ready geometric interpretation. Here we call it the *norm of the expected TD update*:

$$\text{NEU}(\theta) = \mathbb{E}[\delta\phi]^\top \mathbb{E}[\delta\phi] \,. \tag{6}$$

4. Derivation of the new algorithms

In this section we derive two new algorithms as stochastic gradient descent in the projected Bellman error objective (5). We first establish some relationships between the relevant expectations and vector-matrix quantities:

$$\mathbb{E}\big[\phi\phi^\top\big] = \sum_s d_s \phi_s \phi_s^\top = \Phi^\top D \Phi,$$

$$\begin{aligned}\mathbb{E}[\delta\phi] \quad &= \quad \sum_s d_s \phi_s \left(R_s + \gamma \sum_{s'} P_{ss'} V_\theta(s') - V_\theta(s) \right) \\ &= \quad \Phi^\top D (T V_\theta - V_\theta),\end{aligned}$$

and note that

$$\begin{aligned}\Pi^\top D \Pi \quad &= \quad (\Phi(\Phi^\top D \Phi)^{-1} \Phi^\top D)^\top D (\Phi(\Phi^\top D \Phi)^{-1} \Phi^\top D) \\ &= \quad D^\top \Phi (\Phi^\top D \Phi)^{-1} \Phi^\top D \Phi (\Phi^\top D \Phi)^{-1} \Phi^\top D \\ &= \quad D^\top \Phi (\Phi^\top D \Phi)^{-1} \Phi^\top D.\end{aligned}$$

Using these relationships, the projected objective can be written in terms of expectations as

$$\begin{aligned}&\text{MSPBE}(\theta) \\ &= \quad \| V_\theta - \Pi T V_\theta \|^2_D \\ &= \quad \| \Pi(V_\theta - T V_\theta) \|^2_D \\ &= \quad (\Pi(V_\theta - T V_\theta))^\top D (\Pi(V_\theta - T V_\theta)) \\ &= \quad (V_\theta - T V_\theta)^\top \Pi^\top D \Pi (V_\theta - T V_\theta) \\ &= \quad (V_\theta - T V_\theta)^\top D^\top \Phi (\Phi^\top D \Phi)^{-1} \Phi^\top D (V_\theta - T V_\theta) \\ &= \quad (\Phi^\top D (T V_\theta - V_\theta))^\top (\Phi^\top D \Phi)^{-1} \Phi^\top D (T V_\theta - V_\theta) \\ &= \quad \mathbb{E}[\delta\phi]^\top \mathbb{E}\big[\phi\phi^\top\big]^{-1} \mathbb{E}[\delta\phi] \,.\end{aligned}$$

From this form, it is clear that MSPBE differs from NEU only by the inclusion of the inverse of the feature-covariance matrix. Like Sutton, Szepesvari and Maei (2009), we use a second modifiable parameter $w \in \Re^n$ to form a quasi-stationary estimate of all but one of the expectations in the gradient of the objective function, thereby avoiding the need for two independent samples. Here we use a conventional linear predictor which causes w to approximate

$$w \approx \mathbb{E}\big[\phi\phi^\top\big]^{-1}\mathbb{E}[\delta\phi]\,. \tag{7}$$

Using this, we can write the gradient of the MSPBE objective function as

$$\begin{aligned}\frac{1}{2}\nabla\text{MSPBE}(\theta) &= \mathbb{E}\big[(\phi-\gamma\phi')\phi^\top\big]\,\mathbb{E}\big[\phi\phi^\top\big]^{-1}\mathbb{E}[\delta\phi] \\ &\approx \mathbb{E}\big[(\phi-\gamma\phi')\phi^\top\big]\,w,\end{aligned}$$

which can be directly sampled. The resultant $O(n)$ algorithm, which we call GTD-2, is

$$\theta_{k+1} = \theta_k + \alpha_k(\phi_k - \gamma\phi'_k)(\phi_k^\top w_k), \tag{8}$$

where w_k is updated by

$$w_{k+1} = w_k + \beta_k(\delta_k - \phi_k^\top w_k)\phi_k. \tag{9}$$

The derivation of our second new algorithm, which we call *TD(0) with gradient correction*, starts from the same expression for the gradient and then takes a slightly different route:

$$\begin{aligned}&\frac{1}{2}\nabla\text{MSPBE}(\theta) \\ &= \mathbb{E}\big[(\phi-\gamma\phi')\phi^\top\big]\,\mathbb{E}\big[\phi\phi^\top\big]^{-1}\mathbb{E}[\delta\phi] \\ &= \big(\mathbb{E}\big[\phi\phi^\top\big]-\gamma\mathbb{E}\big[\phi'\phi^\top\big]\big)\,\mathbb{E}\big[\phi\phi^\top\big]^{-1}\mathbb{E}[\delta\phi] \\ &= \mathbb{E}[\delta\phi]-\gamma\mathbb{E}\big[\phi'\phi^\top\big]\,\mathbb{E}\big[\phi\phi^\top\big]^{-1}\mathbb{E}[\delta\phi] \\ &\approx \mathbb{E}[\delta\phi]-\gamma\mathbb{E}\big[\phi'\phi^\top\big]\,w,\end{aligned}$$

which is then sampled, resulting in the $O(n)$-computation algorithm:

$$\theta_{k+1} = \theta_k + \alpha_k\delta_k\phi_k - \alpha\gamma\phi'_k(\phi_k^\top w_k), \tag{10}$$

where w_k is generated by (9) as in GTD-2. Note that the update to θ_k is the sum of two terms, and that the first term is exactly the same as the update of conventional linear TD(0) (2). The second term is essentially an adjustment or correction of the TD(0) update so that it follows the gradient of the MSPBE objective function. If the second parameter vector is initialized to $w_0 = 0$ and β_k is small, then this algorithm will start out making nearly the same updates as conventional linear TD(0). Note also that after the convergence of θ_k, w_k will converge to zero again.

5. Proof of convergence of GTD-2

The purpose of this section is to establish that the GTD-2 algorithm converges with probability one to the TD fixed point (4) in the i.i.d. setting under standard assumptions. In particular, we have the following result:

Theorem 1 (Convergence of GTD-2)**.** *Consider the GTD-2 iterations (8) and (9) with step-size sequences α_k and β_k satisfying $\beta_k = \eta\alpha_k$, $\eta > 0$, $\alpha_k, \beta_k \in (0,1]$, $\sum_{k=0}^\infty \alpha_k = \infty$, $\sum_{k=0}^\infty \alpha_k^2 < \infty$. Further assume that (ϕ_k, r_k, ϕ'_k) is an i.i.d. sequence with uniformly bounded second moments. Let $A = \mathbb{E}\big[\phi_k(\phi_k-\gamma\phi'_k)^\top\big]$, $b = \mathbb{E}[r_k\phi_k]$, and $C = \mathbb{E}\big[\phi_k\phi_k^\top\big]$. Assume that A and C are non-singular. Then the parameter vector θ_k converges with probability one to the TD(0) fixed point (4).*

Proof. The proof is very similar to that given by Sutton, Szepesvári and Maei (2009) for GTD, and we refer the reader to that reference for further details. It is shown there that the TD fixed point can be written as the condition

$$-A\theta + b = 0. \tag{11}$$

Thus it suffices for the theorem to show convergence to a solution to (11). The proof is based on the ordinary-differential-equation (ODE) approach (Borkar & Meyn 2000).

First, we rewrite the algorithm's two iterations as a single iteration in a combined parameter vector with $2n$ components, $\rho_k^\top = (d_k^\top, \theta_k^\top)$, where $d_k = w_k/\sqrt{\eta}$, and a new reward-related vector with $2n$ components, $g_{k+1}^\top = (r_k\phi_k^\top, 0^\top)$, as follows:

$$\rho_{k+1} = \rho_k + \alpha_k\sqrt{\eta}\,(G_{k+1}\rho_k + g_{k+1})\,,$$

where

$$G_{k+1} = \begin{pmatrix} -\sqrt{\eta}\phi_k\phi_k^\top & \phi_k(\gamma\phi'_k-\phi_k)^\top \\ (\phi_k-\gamma\phi'_k)\phi_k^\top & 0\end{pmatrix}.$$

Let $G = \mathbb{E}[G_k]$ and $g = \mathbb{E}[g_k]$. Note that G and g are well-defined as by the assumption the process $\{\phi_k, r_k, \phi'_k\}_k$ is i.i.d. In particular,

$$G = \begin{pmatrix} -\sqrt{\eta}\,C & -A \\ A^\top & 0\end{pmatrix}, \quad g = \begin{pmatrix} b \\ 0\end{pmatrix}.$$

Further, note that (11) follows from

$$G\rho + g = 0, \tag{12}$$

where $\rho^\top = (d^\top, \theta^\top)$.

Now we apply Theorem 2.2 of Borkar & Meyn (2000). For this purpose we write $\rho_{k+1} = \rho_k + \alpha_k\sqrt{\eta}(G\rho_k + g + (G_{k+1}-G)\rho_k + (g_{k+1}-g)) = \rho_k + \alpha'_k(h(\rho_k) + M_{k+1})$,

where $\alpha'_k = \alpha_k\sqrt{\eta}$, $h(\rho) = g + G\rho$ and $M_{k+1} = (G_{k+1} - G)\rho_k + g_{k+1} - g$. Let $\mathcal{F}_k = \sigma(\rho_1, M_1, \ldots, \rho_{k-1}, M_k)$ be sigma fields generated by the quantities $\rho_i, M_i, i \le k, k \ge 1$. Theorem 2.2 requires the verification of the following conditions: (i) The function h is Lipschitz and $h_\infty(\rho) = \lim_{r\to\infty} h(r\rho)/r$ is well-defined for every $\rho \in \Re^{2n}$; (ii-a) The sequence $(M_k, \mathcal{F}_k)$ is a martingale difference sequence, and (ii-b) for some $c_0 > 0$, $\mathbb{E}\big[\|M_{k+1}\|^2 \,|\, \mathcal{F}_k\big] \le c_0(1+\|\rho_k\|^2)$ holds for any initial parameter vector ρ_1; (iii) The sequence α'_k satisfies $0 < \alpha'_k \le 1$, $\sum_{k=1}^\infty \alpha'_k = \infty$, $\sum_{k=1}^\infty (\alpha'_k)^2 < +\infty$; (iv) The ODE $\dot\rho = h_\infty(\rho)$ has the origin as a globally asymptotically stable equilibrium and (v) The ODE $\dot\rho = h(\rho)$ has a unique globally asymptotically stable equilibrium. Clearly, $h(\rho)$ is Lipschitz with coefficient $\|G\|$ and $h_\infty(\rho) = G\rho$. By construction, $(M_k, \mathcal{F}_k)$ satisfies $\mathbb{E}[M_{k+1}|\mathcal{F}_k] = 0$ and $M_k \in \mathcal{F}_k$, i.e., it is a martingale difference sequence. Condition (ii-b) can be shown to hold by a simple application of the triangle inequality and the boundedness of the second moments of (ϕ_k, r_k, ϕ'_k). Condition (iii) is satisfied by our conditions on the step-size sequences α_k, β_k.

For the last two conditions, we begin by showing that the real parts of all the eigenvalues of G are negative. First, we show that G is non-singular. Using the determinant rule for partitioned matrices, we get $\det(G) = \det(A^\top C^{-1} A) = (\det C)^{-1}(\det A)^2 \neq 0$. This indicates that all the eigenvalues of G are non-zero. Now, let $\lambda \in \mathbb{C}$, $\lambda \neq 0$ be an eigenvalue of G with corresponding normalized eigenvector $x \in \mathbb{C}^{2n}$; that is, $\|x\|^2 = x^*x = 1$, where x^* is the complex conjugate of x. Hence $x^*Gx = \lambda$. Let $x^\top = (x_1^\top, x_2^\top)$, where $x_1, x_2 \in \mathbb{C}^n$. Using the definition of G, $\lambda = x^*Gx = -\sqrt{\eta}\|x_1\|_C^2 - x_1^* A x_2 + x_2^* A^\top x_1$, where $\|x_1\|_C^2 = x_1^* C x_1$. Because A is real, $A^* = A^\top$, and it follows that $(x_1^* A x_2)^* = x_2^* A^\top x_1$. Thus, $\mathrm{Re}(\lambda) = \mathrm{Re}(x^*Gx) = -\sqrt{\eta}\|x_1\|_C^2 \le 0$. We are now done if we show that x_1 cannot be zero. If $x_1 = 0$, then from $\lambda = x^*Gx$ we get that $\lambda = 0$, which contradicts with $\lambda \neq 0$. Thus, (iv) is satisfied. Finally, for the ODE $\dot\rho = h(\rho)$, note that $\rho^* = -G^{-1}g$ is the unique asymptotically stable equilibrium with $\bar V(\rho) = (G\rho+g)^T(G\rho+g)/2$ as its associated strict Liapunov function. The claim now follows. □

6. Proof of Convergence of TD(0) with Gradient Correction

Theorem 2 (Convergence of TD(0) with Gradient Correction)**.** *Consider the iterations (10) and (9) of the TD(0) with Gradient Corrections algorithm. Let the step-size sequences α_k and β_k satisfy in this case $\alpha_k, \beta_k > 0$, for all k, $\sum_{k=0}^\infty \alpha_k = \sum_{k=0}^\infty \beta_k = \infty$, $\sum_{k=0}^\infty \alpha_k^2$, $\sum_{k=0}^\infty \beta_k^2 < \infty$ and that $\dfrac{\alpha_k}{\beta_k} \to 0$ as $k \to \infty$. Further assume that (ϕ_k, r_k, ϕ'_k) is an i.i.d. sequence with uniformly bounded second moments. Let $A = \mathbb{E}\big[\phi_k(\phi_k - \gamma\phi'_k)^\top\big]$, $b = \mathbb{E}[r_k\phi_k]$, and $C = \mathbb{E}\big[\phi_k\phi_k^\top\big]$. Assume that A and C are non-singular. Then the parameter vector θ_k converges with probability one to the TD(0) fixed point (4).*

Proof. The proof of this theorem relies on a two-timescale difference in the step-size schedules $\{\alpha_k\}$ and $\{\beta_k\}$. In particular, the recursion (9) corresponds to the faster recursion as it is governed by a step-size that converges to zero slower, while (10) is the slower recursion (by an analogous reasoning). We first show that the faster recursion (9) converges for a given parameter update $\theta_k \equiv \theta$. Next, we show that the slower recursion (10) converges.

We begin now with the faster recursion (9). Rewrite (9) as

$$w_{k+1} = w_k + \beta_k E[(\delta_k\phi_k - \phi_k\phi_k^T w_k) \mid \mathcal{F}(k)] + \beta_k M_{k+1},$$

where $M_{k+1} = (\delta_k\phi_k - \phi_k\phi_k^T w_k) - E[(\delta_k\phi_k - \phi_k\phi_k^T w_k) \mid \mathcal{F}(k)]$. Here $\mathcal{F}(k) = \sigma(w_l, \theta_l, l \le k; \phi_s, \phi'_s, r_s, s < k)$, $k \ge 1$. For $\theta_k \equiv \theta$, one can see that $\{M_k\}$ is a martingale difference sequence that satisfies

$$E[\|\ M_{k+1}\ \|^2 |\ \mathcal{F}(k)] \le c_1(1+\ \|\ w_k\ \|^2),\ k \ge 0,$$

for some $c_1 > 0$. The above follows as a consequence of the assumption that r_k, ϕ_k, ϕ'_k all have uniformly bounded second moments.

Consider now the system of ordinary differential equations (ODEs)

$$\dot\theta = 0, \tag{13}$$

$$\dot w = E[\delta\phi] - Cw. \tag{14}$$

Along the faster timescale (the one obtained from $\{\beta_k\}$), we show that the trajectories of the algorithm asymptotically track those of the system of ODEs (13)-(14). Note that the recursion (10) can be rewritten as

$$\theta_{k+1} = \theta_k + \beta_k\left(\frac{\alpha_k}{\beta_k}\left(\delta_k\phi_k - \gamma\phi'_k(\phi_k^\top w_k)\right)\right)$$

$$= \theta_k + \beta_k\xi(k),$$

where $\xi(k) \to 0$ as $k \to \infty$ by the (two-timescale) assumption that $\alpha_k = o(\beta_k)$. The claim for the θ_k-recursion follows from the Hirsch lemma (Theorem 1, pp. 339 of Hirsch, 1989). Thus we can write $\theta_k \equiv \theta$, i.e., consider a time-invariant parameter. Consider the function $h(w) = E[\delta\phi] - Cw$, i.e., the driving vector field of the ODE (14). Now for (14), $w^* = C^{-1}E[\delta\phi]$ can be seen to be the unique globally asymptotically stable equilibrium with $V(w) = (E[\delta\phi] - Cw)^T(E[\delta\phi - Cw)/2$ as an associated strict Liapunov function. (Note again that the quantity δ in w^* or for that matter in the ODE (14) has $\theta_k \equiv \theta$.) Let

$h_\infty(\cdot)$ be the function defined by $h_\infty(w) = \lim_{r\to\infty} \frac{h(rw)}{r}$. Then $h_\infty(w) = -Cw$. For the ODE

$$\dot{w} = h_\infty(w) = -Cw,$$

the origin is a globally asymptotically stable equilibrium because C is positive definite and symmetric. Assumptions (A1) and (A2) of Borkar & Meyn 2000 are now verified and by Theorem 2.2 of Borkar & Meyn 2000, we obtain that $\| w_t - w^* \| \to 0$ almost surely as $t \to \infty$.

Consider now the slower timescale recursion (10). In light of the above, one can rewrite (10) as

$$\theta_{k+1} = \theta_k + \alpha_k(E[\delta_k\phi_k \mid \mathcal{G}(k)]$$

$$-\gamma E[\phi'_k\phi_k^T \mid \mathcal{G}(k)]C^{-1}E[\delta_k\phi_k]) + \alpha_k Z_{k+1},$$

for a suitable Z_{k+1} defined from the above and recursion (10). Here $\mathcal{G}(k) = \sigma(\theta_l, l \le k; \phi_s, \phi'_s, r_s, s < k)$. Then $\{Z_k, \mathcal{G}(k)\}$ can be seen to be martingale-difference sequence that satisfies

$$E[\| Z_{k+1} \|^2 | \mathcal{G}(k)] \le c_2(1 + \| \theta_k \|^2),\ k \ge 0,$$

for some constant $c_2 > 0$. Consider now the following ODE associated with (10):

$$\dot{\theta} = (I - E[\gamma\phi'\phi^T]C^{-1})E[\delta\phi]. \tag{15}$$

Let $\bar{h}(\theta)$ be the driving vector field of the ODE (15). Note that

$$\bar{h}(\theta) = (I - E[\gamma\phi'\phi^T]C^{-1})E[\delta\phi]$$
$$= (C - E[\gamma\phi'\phi^T])C^{-1}E[\delta\phi]$$
$$= (E[\phi\phi^T] - E[\gamma\phi'\phi^T])C^{-1}E[\delta\phi]$$
$$A^TC^{-1}(-A\theta + b),$$

since $E[\delta\phi] = -A\theta + b$.

Now $\theta^* = A^{-1}b$ can be seen to be the unique globally asymptotically stable equilibrium for (15) with $W(\theta) = (A^TC^{-1}(-A\theta+b))^T(A^TC^{-1}(-A\theta+b))/2$ as an associated strict Liapunov function. Now let $\bar{h}_\infty(\theta) = \frac{\bar{h}(r\theta)}{r}$. In this case, $\bar{h}_\infty(\theta) = -A^TC^{-1}A\theta$. Consider now the ODE

$$\dot{\theta} = -A^TC^{-1}A\theta.$$

Because C^{-1} is positive definite and symmetric and A has full rank (as it is nonsingular by assumption), the matrix $A^TC^{-1}A$ is also positive definite and symmetric. The above ODE (corresponding to the ∞-system) has the origin as its unique globally asymptotically stable equilibrium. The assumptions (A1)-(A2) of Borkar & Meyn 2000 are once again verified and the claim follows. □

7. Empirical Results

In order to give a flavor for the performance of our new algorithms, we use two tasks. The first is a random walk task with 7 states. The end states are absorbing; the middle states transition uniformly at random to one of the neighbors. The rewards are zero everywhere, except at the rightmost state, where the reward is $+1$. The discount factor is $\gamma = 1$. We use three different feature representations. The first is the familiar table lookup. The second, which we call "inverted features", uses feature vectors which are all $1/2$, except for the bit corresponding to the state, which is set to zero. The value $1/2$ is chosen to give the feature vectors norm 1. In this case, the value function can be represented exactly, but the features are very correlated, which makes the learning problem very difficult. The third uses a symmetric function approximator with 3 features, covering states (2,3), (3,4,5) and (5,6) respectively. The values of the features are $1/\sqrt{(2)}$, so that their norm is 1. In this case, the exact value function cannot be represented anymore. The second task is Boyan's chain example, which has been used in many comparisons of TD-style algorithms. This task has 14 states, and 4 features are used to represent the value function. However, the exact value can be represented using the given features. This is an episodic problem, with $\gamma = 1$; the exact description can be found in (Boyan, 2002).

The initial values for the parameters were 0 for all algorithms. For the random walk task, we used learning rate values $\alpha = 2^{-n}$ with n from 1 to 5, for the tabular and inverted representations, and $n = 1 \ldots 7$ for the function approximation. For TD(0) with gradient corrections (TDC), GTD and GTD-2, the ratio $\eta = \beta/\alpha$ took values from the set $\{1/4, 1/2, 1, 2\}$. For the Boyan chain example, we used $\eta \in 2^n$ where $n = -4 \ldots 1$. For α, we used the same values as in the previous experiments; however, GTD's behavior was optimal at values of $\alpha > 1$, so we include these here as well.

In all the experiments, we performed 100 independent runs. The standard error bars are on the order of 10^{-4}, so are not included. Figure 3 summarizes the results. For the plots presented here, we use the root of the MSPBE (as defined in Section 4) as the performance measure. However, we computed also the root mean squared prediction error (compared to the true value function) and the root NEU (see Equation 6). In all cases, these measures are consistent with each other, and the ranking of the algorithms is the same for all measures. For each task, the left graph shows the area under the learning curve, over the range of α, for different values of β. The right graph shows learning curves for each algorithm at its best parameter settings (as determined from the previous parameter study.

Both TD(0) with gradient corrections and GTD-2 are sig-

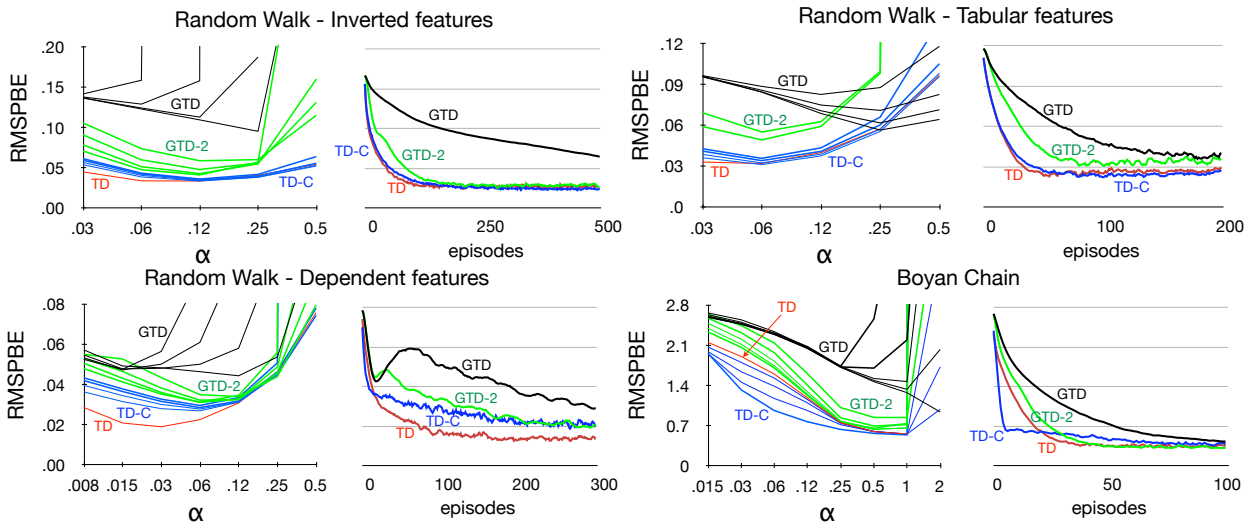

Figure 3. Empirical results for random walk and Boyan chain

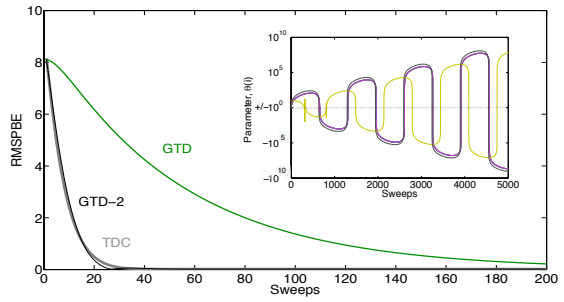

Figure 4. Baird's counterexample: Gradient algorithms converge, while TD(0) diverges

nificantly faster than GTD for all tasks. Moreover, TDC is comparable in speed to TD(0) for a range of values of the parameter η.

However, unlike TD(0), gradient-based algorithms have the advantage of convergence in the case of off-policy learning. To illustrate this, we use the standard counterexample proposed by Baird (1999). For GTD-2 and TDC we used $\alpha = 0.05$, $\beta = 0.5$ and for GTD, $\alpha = 0.005$ and $\beta = 0.2$. Figure 4 shows the familiar oscillation of the value function parameters with TD(0). The second plot shows that, as expected, all gradient algorithms converge, and the projected Bellman error goes to 0. GTD-2 and TDC are faster than GTD in this example as well.

8. Conclusions and future work

We presented two new classes of gradient-based temporal-difference learning algorithms, which minimize a natural performance measure, the projected Bellman error. We proved convergence of both algorithms with off-policy learning and linear function approximation. Both algorithms have linear complexity in the number of features used in the function approximation. Both are significantly faster than GTD, a previously proposed gradient-based TD algorithm. Moreover, the TD(0) algorithm with gradient corrections is comparable in speed to TD(0). This is the first time that all these desirable features (linear complexity, speed and convergence with off-policy learning and function approximation) are achieved. Future work includes extensions of these methods to eligibility traces and on-trajectory (rather than i.i.d) sampling. Gathering more empirical experience with these methods would also be beneficial.

References

Baird, L. C. (1995). Residual algorithms: Reinforcement learning with function approximation. In *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 30–37. Morgan Kaufmann.

Baird, L. C. (1999). Reinforcement Learning Through Gradient Descent. PhD thesis, Carnegie-Mellon University Technical Report CMU-CS-99-132.

Barnard, E. (1993). Temporal-difference methods and Markov models. *IEEE Transactions on Systems, Man, and Cybernetics 23*(2):357–365.

Borkar, V. S. and Meyn, S. P. (2000). The ODE method for convergence of stochastic approximation and rein-

forcement learning. *SIAM Journal on Control And Optimization 38*(2):447–469.

Boyan, J. (2002). Technical update: Least-squares temporal difference learning. *Machine Learning 49*:233–246.

Bradtke, S., Barto, A. G. (1996). Linear least-squares algorithms for temporal difference learning. *Machine Learning 22*:33–57.

Dayan, P. (1992). The convergence of TD(λ) for general λ. *Machine Learning 8*:341–362.

Geramifard, A., Bowling, M., Sutton, R. S. (2006). Incremental least-square temporal difference learning. *Proceedings of the National Conference on Artificial Intelligence*, pp. 356–361.

Hirsch, M. W. (1989). Convergent activation dynamics in continuous time networks. *Neural Networks 2*:331–349.

Precup, D., Sutton, R. S. and Dasgupta, S. (2001). Off-policy temporal-difference learning with function approximation. *Proceedings of the 18th International Conference on Machine Learning*, pp. 417–424.

Precup, D., Sutton, R. S., Paduraru, C., Koop, A., Singh, S. (2006). Off-policy Learning with Recognizers. *Advances in Neural Information Processing Systems 18*.

Silver, D., Sutton, R. S., Müller, M. (2007). Reinforcement learning of local shape in the game of Go. *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pp. 1053–1058.

Sturtevant, N. R., White, A. M. (2006). Feature construction for reinforcement learning in hearts. In *Proceedings of the 5th International Conference on Computers and Games*.

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

Sutton, R. S., Precup D., and Singh, S (1999). Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence 112*:181–211.

Sutton, R. S., Szepesvári, Cs., Maei, H. R. (2009). A convergent O(n) algorithm for off-policy temporal-difference learning with linear function approximation. In *Advances in Neural Information Processing Systems 21*. MIT Press.

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