Technical Report CS-2012-01

L_1 Regularized Linear Temporal Difference Learning

Christopher Painter-Wakefield and Ronald Parr
Department of Computer Science
Duke University
Durham, NC
paint007@cs.duke.edu, parr@cs.duke.edu

January 6, 2012

Abstract

Several recent efforts in the field of reinforcement learning have focused attention on the importance of regularization, but the techniques for incorporating regularization into reinforcement learning algorithms, and the effects of these changes upon the convergence of these algorithms, are ongoing areas of research. In particular, little has been written about the use of regularization in online reinforcement learning. In this paper, we describe a novel online stochastic approximation algorithm for reinforcement learning. We prove convergence of the online algorithm and show that the L_1 regularized linear fixed point of LARS-TD and LC-TD is an equilibrium fixed point of the algorithm.

1 Introduction

The importance of regularization in regression problems is well established, with the sparsity inducing properties of L_1 regularization receiving particular interest and attention of late [6, 9, 16, 20]. Recent developments in reinforcement learning have also focused attention on the issue of regularization. Kolter and Ng [13] introduced the notion of an L_1 *regularized linear fixed point*, and the LARS-TD algorithm for finding it; their experiments demonstrated the effectiveness of this technique at selecting good features in the presence of noise features that could otherwise cause overfitting. Johns et al. [12] produced the algorithm LC-TD, which also finds the L_1 regularized linear fixed point, and used it in a modified policy iteration setting to find (approximate) optimal policies.

LARS-TD and LC-TD are inherently batch learning methods; regularization in online reinforcement learning is a topic heretofore unexplored. One reason for this is that regularization, particularly L_1 regularization, is most useful in the low data regime, where noise has the most impact and overfitting is most likely. A typical assumption in online learning is that data are infinite, or at least plentiful. This may not always be the case, of course, and furthermore it is occasionally desirable to apply on "online" algorithm to batch data, in which case overfitting may indeed be a concern.

An alternative motivation for using L_1 regularization in online learning is in reducing the memory and computational requirements of the online updates. Langford et al. [14] describe in some detail an algorithm which utilizes a class of shrinkage operators (including the L_1 regularizing soft-threshold shrinkage operator) in online supervised learning. In this work, sparsity of the iterates and sparsity in the features are leveraged to

speed up the algorithm when working with large data sets. Sparsity in the iterates allows the algorithm to keep in memory only those features which are currently non-zero weighted, preventing additional costs due to swapping. Sparsity in the features directly aids computational efficiency, as only weights of non-zero features need be updated. The Langford et al. approach translates directly into the RL setting, given a problem with a large set of sparse features such that the non-zero features can be determined efficiently from the state alone, without iterating over all features. Certain types of sparse coding such as CMACs fall naturally into the category of sparse features for which the non-zero features are determinable from the state alone.

In this paper we present an L_1 regularized, online stochastic approximation algorithm, based on an iterative update equation sharing the fixed point of LARS-TD and LC-TD. We validate the ability of the online algorithm to converge to the fixed point described by theory in two sample problems, and demonstrate the tradeoff between sparsity and accuracy in a third problem.

2 Background and Notation

In this section we review Markov reward processes and regression, and we define the L_1 regularized linear fixed point and review the LARS-TD [13] algorithm.

2.1 Markov Reward Processes

This work aims to discover optimal, or near-optimal, values functions for Markov reward processes (MRPs): $M = (S, P, R, \gamma)$. Given a state $s \in S$, the probability of a transition to a state $s' \in S$ is given by $P(s'|s)$, and results in an expected reward of $R(s)$. We do not address the question of optimizing the policy for a Markov Decision Process, though we note that policy evaluation, where $P = P_\pi$, by some policy π, is an important intermediate step in many algorithms. A discount factor γ discounts future rewards such that the present value of a trajectory $s_{t=0} \dots s_{t=n}$ is $\sum_{t=0}^{n} \gamma^t R(s_t)$.

The value function V over states satisfies the *Bellman equation*:

$$V = TV = R + \gamma PV,$$

where T is the *Bellman operator* and V is the fixed point of this operator.

In practice, the value function, the transition model, and the reward function are often too large to permit an explicit, exact representation. In such cases, an approximation architecture is used for the value function. A common choice is $\hat{V} = \Phi w$, where w is a vector of k scalar weights and Φ stores a set of k features in an $n \times k$ feature matrix. Since n is often intractably large, Φ can be thought of as populated by k linearly independent *basis functions*, $\varphi_1 \dots \varphi_k$, which define the columns of Φ.

For the purposes of estimating w, it is common to replace Φ with $\hat{\Phi}$, which samples rows of Φ, though for conciseness of presentation we will use Φ for both, since algorithms for estimating w are essentially identical if $\hat{\Phi}$ is substituted for Φ. Typical linear function approximation algorithms [4] solve for the w which is a fixed point:

$$\Phi w = \Pi^\sigma (R + \gamma \Phi' w) = \Pi^\sigma T \Phi w, \tag{1}$$

where Π^σ is the σ-weighted L_2 projection and where Φ' is $P\Phi$ in the explicit case and composed of sampled next features in the sampled case. We likewise overload T for the sampled case.

2.2 L_1 Regularized Regression

Least squares regression seeks to minimize the weighted sum of squared errors between the predictor and the target x:

$$w = \arg\min_u \frac{1}{2}\|\Phi u - x\|_\sigma^2,$$

where $\|\cdot\|_\sigma$ is the σ-weighted L_2 norm. This classic approach is well known to have the solution:

$$\Phi w = \Pi^\sigma x$$

where Π^σ is the σ-weighted L_2 projection. Since regression can easily overfit noise in x when an expressive basis is used, many approaches to regularizing the classic solution have been proposed.

L_1 regularized regression has recently been the focus of a great deal of research. The *Lasso* [20] is the L_1 constrained regression problem

$$w = \arg\min_u \frac{1}{2}\|\Phi u - x\|_\sigma^2 \text{ subject to } \|u\|_1 \leq \lambda.$$

The Lasso problem can be solved directly using linear programming, but this approach is not very efficient for large problems. This motivates the LARS algorithm [9], which can be thought of as a homotopy method for solving the Lasso, as well as various gradient descent [8, 11, 14] and pursuit [5] algorithms. Conveniently, these approaches are more readily adapted to reinforcement learning than the direct, linear programming implementation of the Lasso.

LARS and other methods formulate L_1 regularization using unconstrained minimization:

$$w = \arg\min_u \frac{1}{2}\|\Phi u - x\|_\sigma^2 + \beta\|u\|_1. \tag{2}$$

It is a straightforward application of Lagrange multiplier theory to show that the optimum of an instance of the above matches a solution to a Lasso instance with the same objective term, and that furthermore there is a continuous mapping between λ and β preserving equivalence.

Equation (2) can be expressed equivalently as $\Phi w = \Pi^\sigma_{\{1,\beta\}} x$ where $\Pi^\sigma_{\{1,\beta\}}$ is the L_1 *regularized least-squares projection operator* for some $\beta > 0$.

2.3 L_1 Regularized Linear Fixed Point

Kolter and Ng [13] introduced the problem

$$w = \arg\min_u \frac{1}{2}\|\Phi u - T\Phi w\|_\sigma^2 + \beta\|u\|_1, \tag{3}$$

which can be rewritten as the L_1 *regularized linear fixed point* problem

$$\Phi w = \Pi^\sigma_{\{1,\beta\}} T\Phi w.$$

The above is simply the linear fixed point of equation (1) with the addition of L_1 regularization. Ghavamzadeh et al. [10] analyze a generalization of the operator $\Pi^\sigma_{\{1,\beta\}} T$, which they show to be a contraction, ensuring the existence and uniqueness of the fixed point Φw (note that w by itself is not required to be unique).

Algorithms which solve for the L_1 regularized linear fixed point include LARS-TD [13] and LC-TD [12]. LARS-TD follows a homotopy method, in which the regularization parameter is slowly shrunken to the

desired value, maintaining at all times a set of fixed point criteria with respect to the regularization parameter. LC-TD [12] restates the same fixed point criteria as a *linear complementarity problem* (LCP), which can be solved by a variety of existing solvers. As the fixed point criteria are also critical to understanding the fixed point behavior of our online algorithm, we review their derivation here.

The fixed point conditions are obtained by first taking the subdifferential of $L(u) = \frac{1}{2}\|\Phi u - T\Phi w\|_\sigma^2 + \beta\|u\|_1$:

$$\nabla L = \Phi^T \Sigma(\Phi u - T\Phi w) + \beta\, \mathrm{SGN}(u)$$

where $\mathrm{SGN}(u)$ is a set-valued function defined componentwise as

$$\mathrm{SGN}(u)_i = \begin{cases} \{+1\}, & u_i > 0 \\ [-1, 1], & u_i = 0 \\ \{-1\}, & u_i < 0, \end{cases}$$

and Σ is the diagonal matrix with $diag(\Sigma) = \sigma$, the stationary distribution.[1]

Setting the subdifferential to zero (since the subdifferential is set-valued, the actual requirement is that zero is in the subdifferential) yields the optimality conditions

$$[\Phi^T \Sigma(\Phi u - T\Phi w)]_i \begin{cases} = -\beta, & u_i > 0 \\ \in [-\beta, \beta], & u_i = 0 \\ = +\beta, & u_i < 0. \end{cases}$$

At the fixed point we require that $u = w$, thus the fixed point is characterized by the above conditions with u replaced everywhere by w:

$$[\Phi^T \Sigma(\Phi w - T\Phi w)]_i \begin{cases} = -\beta, & w_i > 0 \\ \in [-\beta, \beta], & w_i = 0 \\ = +\beta, & w_i < 0. \end{cases} \tag{4}$$

Any vector w satisfying these conditions minimizes the right-hand side of equation (3), yielding the fixed point Φw.

3 Regularized Linear TD

In this section we develop and analyze a new, iterative algorithm for finding the regularized linear fixed point. LARS-TD and LC-TD are inherently *batch* methods; samples are obtained prior to running the algorithm, and each iteration generates an update based on the entire sample corpus. Our new algorithm can be used with batch data but also permits *online* learning of the value function.

3.1 Linear TD as Gradient Descent

Our new algorithm can be motivated as a gradient descent algorithm in the spirit of linear temporal difference learning (TD) [18]. Linear TD can be viewed, somewhat loosely, as a gradient descent method minimizing the "objective" function $\frac{1}{2}\|\Phi u - T\Phi w\|^2$. The linear TD update is derived from the standard gradient learning

[1] Kolter and Ng [13] and Johns et al. [12] omit Σ. Here we explicitly weight by the stationary distribution to ensure consistency between LARS-TD and our online algorithm described in section 3. For N samples, Σ can be replaced by 1/N.

rule by taking the gradient of the objective function with respect to u, together with the constraint $u = w$:

$$\begin{aligned} w_{t+1} &:= w_t - \alpha_t \nabla_u \left(\frac{1}{2} \|\Phi u - T\Phi w_t\|_\sigma^2 \right) \\ &:= w_t + \alpha_t \nabla_u (\Phi u - T\Phi w_t) \cdot \Sigma (T\Phi w_t - \Phi u) \\ &:= w_t + \alpha_t \, \Phi^T \Sigma (T\Phi w_t - \Phi w_t) \end{aligned} \tag{5}$$

Here α_t is a (possibly time-dependent) learning rate parameter. In online learning, the values of Φw and $T\Phi w$ are replaced by samples, yielding a stochastic estimate of the gradient; also, Σ is implicit in the online sampling and no longer appears in the update.

3.2 Linear TD with Soft-Thresholding

We now turn to a class of gradient descent algorithms which use thresholding to effect L_1 regularization. Introduced by Donoho and Johnstone [7], the *soft-threshold shrinkage operator* Ψ_ν is defined as

$$\Psi_\nu(x) = \mathrm{sgn}(x) \odot \max\{|x| - \nu, 0\} \tag{6}$$

where $\nu > 0$. The sgn and $\max$ operations are componentwise, and $\odot$ signifies componentwise multiplication. In words, the shrinkage operator reduces the magnitude of each element of x by ν, truncating to zero if the magnitude was already $\leq \nu$.

A number of authors [6, 8, 11, 14] make use of this operator composed with gradient descent in an iterative fixed point method for solving the regression problem (2):

$$w_{t+1} := \Psi_{\alpha\beta}(w_t - \alpha \, \Phi^T(\Phi w_t - x)), \tag{7}$$

where α is a (typically small) positive step size. Duchi and Singer [8] establish convergence (under mild conditions) for the online variant, replacing the gradient $\Phi^T(\Phi w - x)$ with a stochastic estimate.

An important fact for our purposes is that the soft-threshold shrinkage operator is non-expansive in p-norm [11, Lemma 3.2].

In this section we introduce the most straightforward application of the soft-threshold shrinkage operator in the context of linear TD, and show that the fixed point of the resulting iteration satisfies the LARS-TD fixed point conditions (4). If we compose the soft-threshold shrinkage operator with the linear TD gradient update (5) we obtain

$$w_{t+1} := \Psi_{\alpha_t \beta}(w_t + \alpha_t \, \Phi^T \Sigma (T\Phi w_t - \Phi w_t)). \tag{8}$$

This update generates the algorithm **L1TD**, shown in algorithm 1.

We give the following lemma concerning the fixed point of (8); the subsequent proof is nearly identical to the proof given by Hale et al. [11, Proposition 3.1] in the regression setting:

Lemma 1 *A vector w^* is a solution to the LARS-TD fixed point problem (3) with regularization parameter β if and only if, for any $\eta > 0$,*

$$w^* = \Psi_{\eta\beta}(w^* + \eta \Phi^T \Sigma (T\Phi w^* - \Phi w^*)). \tag{9}$$

PROOF Recall that fixed points of LARS-TD are completely characterized by the conditions given in equation (4). Consider the i^{th} element $w^*[i]$, and let $g[i] = [\Phi^T \Sigma (T\Phi w^* - \Phi w^*)]_i$, the i^{th} element of the "gradient". By equations (9) and (6) we have

$$w^*[i] = \mathrm{sgn}(w^*[i] + \eta g[i]) \max\{|w^*[i] + \eta g[i]| - \eta\beta, 0\}. \tag{10}$$

Algorithm 1 L1TD

Input: φ, α, β.
Output: Approximation weights w.
Initialize $w = 0, t = 0$.
repeat
  Obtain samples s, r, s'
  $A \leftarrow \varphi(s)(\varphi(s')^T - \varphi(s)^T))$
  $b \leftarrow \varphi(s)r$
  $w \leftarrow \Psi_{\alpha(t)\beta}(w + \alpha(t)(Aw + b))$.
until convergence.

Algorithm 2 L1TDAlt

Input: $\varphi, \alpha, \beta, \eta$.
Output: Approximation weights w.
Initialize $v = 0, w = 0, t = 0$.
repeat
  Obtain samples s, r, s'
  $A \leftarrow \varphi(s)(\varphi(s')^T - \varphi(s)^T))$
  $b \leftarrow \varphi(s)r$
  $v \leftarrow v + \alpha(t)(w + \eta(Aw + b) - v)$
  $w \leftarrow \Psi_{\eta\beta}(v)$.
until convergence.

Going forward, the $\max$ expression in (10) is nonnegative, and therefore $w^*[i] \neq 0$ implies $\mathrm{sgn}(w^*[i] + \eta g[i]) = \mathrm{sgn}(w^*[i])$. If $w^*[i] > 0$, then

$$\begin{aligned} w^*[i] &= 1(w^*[i] + \eta g[i] - \eta\beta) \\ g[i] &= \beta, \end{aligned}$$

as required by the LARS-TD fixed point conditions.

On the other hand, if $w^*[i] > 0$ and $g[i] = \beta$,

$$\begin{aligned} & \mathrm{sgn}(w^*[i] + \eta g[i]) \max\{|w^*[i] + \eta g[i]| - \eta\beta, 0\} \\ = \; & \mathrm{sgn}(w^*[i] + \eta\beta)(w^*[i] + \eta\beta - \eta\beta) \\ = \; & w^*[i], \end{aligned}$$

as required by equation (10).

A similar argument shows that, if $w^*[i] < 0$, equation (10) implies $g[i] = -\beta$, and $g[i] = -\beta$ implies equation (10).

Finally, if $w^*[i] = 0$, either $g[i] = 0$ or $\mathrm{sgn}(w^*[i] + \eta g[i]) = \pm 1$ and therefore equation (10) is satisfied if and only if

$$\begin{aligned} \max\{|w^*[i] + \eta g[i]| - \eta\beta, 0\} &= 0 \\ |0 + \eta g[i]| - \eta\beta &\leq 0 \\ |g[i]| &\leq \beta \\ g[i] &\in [-\beta, \beta]. \end{aligned}$$

□

3.3 A Modification

The L1TD update (8) is instantly recognizable as an amalgam of linear TD with soft-thresholding, and its form is convenient for proving equivalence with the L_1-regularized linear fixed point (3), but it turns out not to be particularly amenable for a proof of convergence. In this section, therefore, we introduce a slightly altered update, which is easily shown to have the same fixed point as the above, and for which we prove convergence. While we have no formal proof of convergence of (8), its empirical behavior (see section 4) closely matches that of the modified update, and its common pedigree with the modified update is a strong argument for convergence.

A first step in developing our alternate algorithm is to decompose the update into its two basic operations, and reorder them to produce a "half-phase" update. We introduce a new variable, v, with $w_t = \Psi_{\eta\beta}(v_t)$, and we fix $\eta > 0$ at some constant value. The half-phase update equation generating the sequence v is

$$\begin{aligned} v_{t+1} &:= w_t + \eta\Phi^T\Sigma(T\Phi w_t - \Phi w_t) \\ &= \Psi_{\eta\beta}(v_t) + \eta\Phi^T\Sigma(T\Phi\Psi_{\eta\beta}(v_t) - \Phi\Psi_{\eta\beta}(v_t)), \\ &= H(v_t). \end{aligned} \tag{11}$$

Here we take advantage of the fact that the new sequence is dependent solely on v to define a new fixed point operator H. We can recover the original sequence w by writing $w_{t+1} := \Psi_{\eta\beta}(v_{t+1})$ and expanding v_{t+1} by equation (11). We denote the fixed point of the new sequence v^*, and note that it is related to w^* by the mappings $w^* = \Psi_{\eta\beta}(v^*)$, and $v^* = w^* + \eta\Phi^T\Sigma(T\Phi w^* - \Phi w^*)$.

Now we introduce the new iteration, which is more "conservative" in its update, mixing the previous solution with the update due to H according to a time-dependent step size α:

$$v_{t+1} := v_t + \alpha_t(H(v_t) - v_t). \tag{12}$$

This update gives rise to the algorithm **L1TDAlt** given in Algorithm 2. Note that the fixed point v^* of the modified iteration is characterized by $H(v^*) - v^* = 0$, that is, at the fixed point of H. Thus fixed points of the L1TD iteration are equivalent to fixed points of the L1TDAlt iteration via a simple transform.

3.4 Convergence

We now turn to the question of convergence. In particular, we are interested in convergence of the sequence v in the online setting, in which the operator H is replaced by noisy samples of H. Given sample transitions (s_t, r_t, s'_t) and a set of basis functions $\{\varphi_1\ \varphi_2\ \ldots\ \varphi_k\}$, the algorithm performs the updates

$$\begin{aligned} v_{t+1} &:= v_t + \alpha_t(w_t + \eta\varphi(s_t)(r_t + \gamma\varphi(s'_t)^T w_t - \varphi(s)^T w_t) - v_t) \\ w_{t+1} &:= \Psi_{\eta\beta}(v_{t+1}). \end{aligned} \tag{13}$$

Our convergence result depends on the following assumptions:

Assumption 1 *The Markov reward process $M = (S, P, R, \gamma)$ is finite and mixing, with stationary distribution σ.*

Assumption 2 *The sequence (s_t, r_t, s'_t) is sampled i.i.d. from the stationary distribution, σ, of M. That is, $s_t \sim \sigma$, $r_t = R(s_t)$, and s'_t is obtained by making a stochastic transition in accordance with P.*

Assumption 3 *The columns of the matrix Φ comprise a linearly independent set of basis functions evaluated at all states in S. In particular, this implies that Φ is full rank.*

Assumption 4 *The sequence α_t is predetermined and non-increasing, with $\sum_{t=0}^{\infty}\alpha_t = \infty$ and $\sum_{t=0}^{\infty}\alpha_t^2 < \infty$.*

We now give the following theorem regarding convergence:

Theorem 1 *Consider the sequence v_t as defined in (13). Then, under assumptions 1 - 4, v_t converges almost surely as $t \to \infty$.*

PROOF We first restate the sequence v using additional terms. Given sample transitions (s_t, r_t, s'_t) and a set of basis functions $\{\varphi_1\ \varphi_2\ \ldots\ \varphi_k\}$, the algorithm performs the updates

$$v_{t+1} := v_t + \alpha_t(\Psi_{\eta\beta}(v_t) + \eta(A(X_t)\Psi_{\eta\beta}(v_t) + b(X_t)) - v_t), \tag{14}$$

where we define

$$\begin{aligned} X_t &= \{\varphi_t, r_t, \varphi_{t+1}\}, \\ A(X_t) &= \varphi_t(\gamma\varphi_{t+1} - \varphi_t)^T, \text{ and} \\ b(X_t) &= \varphi_t r_t. \end{aligned}$$

We define

$$\begin{aligned} A &= \Phi^T\Sigma(\gamma P\Phi - \Phi); \\ b &= \Phi^T\Sigma R. \end{aligned}$$

and note that

$$E_\sigma[A(X_t)] = \Phi^T\Sigma(\gamma P\Phi - \Phi) = A, \tag{15}$$

$$\text{and} \qquad E_\sigma[b(X_t)] = \Phi^T\Sigma R = b. \tag{16}$$

Here $E_\sigma[\cdot]$ denotes the expectation with respect to the stationary distribution of the Markov process.

Our proof will follow the ODE method as outlined by Borkar [2]. For this purpose, we will rewrite our update in the form

$$v_{t+1} := v_t + \alpha_t[h(v_t) + M_{t+1}],$$

and demonstrate that h and M satisfy the assumptions needed to apply theorem 2 of Borkar [2, p. 15].

Let:

$$\begin{aligned} h(v) &= H(v) - v \\ &= \Psi_{\eta\beta}(v) + \eta(A\Psi_{\eta\beta}(v) + b) - v, \\ M_{t+1} &= \Psi_{\eta\beta}(v) + \eta(A(X_t)\Psi_{\eta\beta}(v) + b(X_t)) - v - (H(v) - v) \\ &= \eta[(A(X_t) - A)\Psi_{\eta\beta}(v_t) + b(X_t) - b] \end{aligned}$$

We now need to satisfy assumptions A1–A3 and A5 of Borkar [2]: we must show (a) that the function h is Lipschitz, (b) that the function $h_\infty = \lim_{r\to\infty} h(rx)/r$ exists, and (c) that the origin is an asymptotically stable equilibrium for the ODE $\dot{v}(t) = h_\infty(v(t))$. We must further show (d) that $\{M_t; \mathcal{F}_t\}$ where $\mathcal{F}_t = (M_0, v_0, M_1, v_1, \ldots M_t, v_t)$ is a martingale difference sequence, that is, that the expectation of M_{t+1} given $\mathcal{F}_t$ is zero, and (e) that, for some $C_0 < \infty$ and any initial condition v_0, $E[\|M_{t+1}\|^2|\mathcal{F}_t] \le C_0(1 + \|v_t\|^2)$.

For (a), to show Lipschitz continuity of h we need C such that $\|h(v) - h(v')\| \le C\|v - v'\|, \forall v, v'$. We have

$$\begin{aligned} & \|h(v) - h(v')\| \\ = \ & \|\Psi_{\eta\beta}(v) + \eta(A\Psi_{\eta\beta}(v) + b) - v \\ & \qquad - (\Psi_{\eta\beta}(v') + \eta(A\Psi_{\eta\beta}(v') + b) - v')\| \\ = \ & \|(I + \eta A)(\Psi_{\eta\beta}(v) - \Psi_{\eta\beta}(v')) + (v' - v)\| \\ \le \ & \|(I + \eta A)\|\|\Psi_{\eta\beta}(v) - \Psi_{\eta\beta}(v')\| + \|v' - v\| \\ \le \ & \|(2I + \eta A)\|\|v - v'\|. \end{aligned}$$

The function h is thus Lipschitz with constant $C = \|2I + \eta A\|$. Here we make use of the triangle inequality, the non-expansiveness of Ψ, and the Cauchy-Schwarz inequality as generalized to matrix norms.

For (b), we have

$$\begin{aligned} h_\infty(v) &= \lim_{r\to\infty} h(rx)/r \\ &= \lim_{r\to\infty} [\Psi_{\eta\beta}(rv) + \eta(A\Psi_{\eta\beta}(rv) + b) - rv]/r \\ &= \lim_{r\to\infty} [(I + \eta A)(rv - \eta\beta \operatorname{sgn}(v)) + \eta b - rv]/r \\ &= \lim_{r\to\infty} [(I + \eta A)(rv - \eta\beta \operatorname{sgn}(v)) - rv]/r \\ &= (I + \eta A)v - v \\ &= \eta A v. \end{aligned}$$

The third step above follows from the fact that all non-zero elements of v are shrunk by no more than $\eta\beta$; as r grows large, all non-zero elements of v are shrunk by this amount, while all zero elements are shrunk by zero. Now (c) follows directly from standard ODE theory and the fact that A is negative definite (by assumption 3 and Lemma 6.6 of Bertsekas and Tsitsiklis [1, p. 300]).

For (d), we note that under assumption 2, our X_t are sampled according to the stationary distribution, and are thus independent of $\mathcal{F}_t$. Using the definition of M and equations (15, 16) we obtain $E[M_{t+1}|\mathcal{F}_t] = \eta((E_\sigma[A(X_t)] - A)\Psi_{\eta\beta}(v_t) + E_\sigma[b(X_t)] - b) = 0$, as required.

Finally, for (e) we note that, given the finiteness assumption on our MRP, the second moments of the reward and our basis functions are finitely bounded; by extension, if we consider M_{t+1} as a linear function of $\Psi_{\eta\beta}(v_t)$, its coefficients are bounded, and its square is a quadratic function of $\Psi_{\eta\beta}(v_t)$ with bounded coefficients. The result follows trivially from the triangle inequality and the non-expansiveness of Ψ. □

Our theorem states convergence in the sense of Theorem 2 of Borkar [2], under which convergence may be to a local optimum. While we do not yet have a proof that convergence is to the fixed point, our experiments show fixed point convergence in every case.

Also, assumption 2 would normally be violated in the general online setting. We make this assumption in order to fit our update to the structure of the proof template given by Borkar [2]. The machinery needed to cope with Markov noise in the online setting is somewhat involved; one approach can be found in section 4.4 of Bertsekas and Tsitsiklis [1].

4 Experiments

Our proofs assume a decaying step size but we use a constant step size in our experiments[2]. We also allowed linearly dependent features in our experiments and considered episodic tasks. We found that these issues did not pose a problem, and expect that theory can be extended to these cases.

We performed experiments on two benchmark problems, Blackjack and Mountain Car, to show the convergence of the online algorithm. We also performed experiments on the Inverted Pendulum problem to illuminate the tradeoff between approximation performance and sparsity. We used both algorithm 1 (L1TD) and algorithm 2 (L1TDAlt) in our experiments; the results were essentially identical between the two algorithms in all cases, and for this reason we give results for only one algorithm in most cases.

[2]Constant step sizes are commonly used to speed up improvement in the value function, possibly at the expense of some oscillation around the true solution. This practice preceded its theoretical justification by Borkar and Meyn [3]. We defer extending their approach to the regularized case for future work.

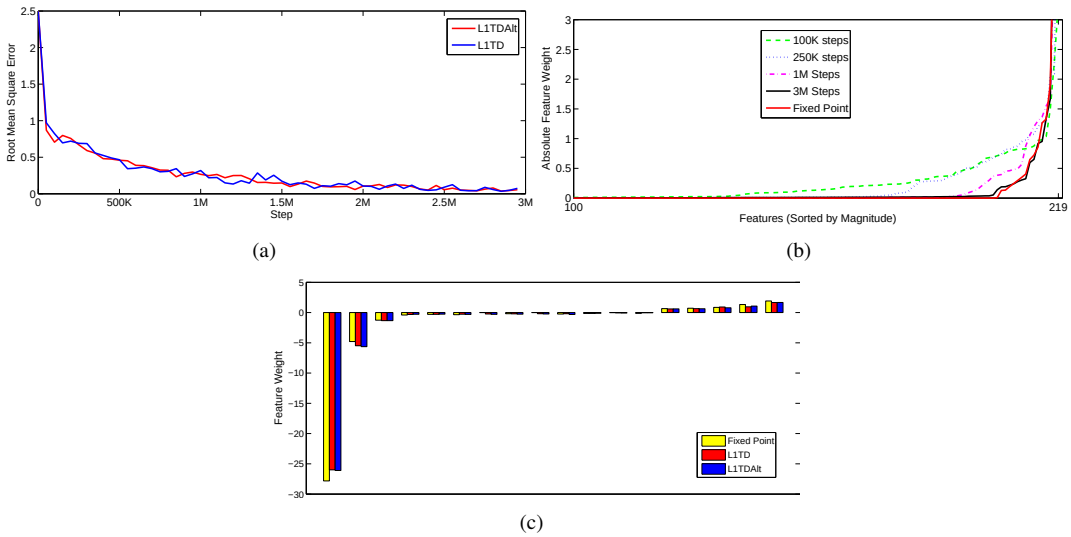


Figure 1: Blackjack: (a) σ-weighted root mean square error between online solutions and the fixed point value function; (b) Sorted absolute weights, L1TD vs. fixed point; (c) Comparison of weights assigned to the top 18 features by L1TD, L1TDAlt, and the fixed point.

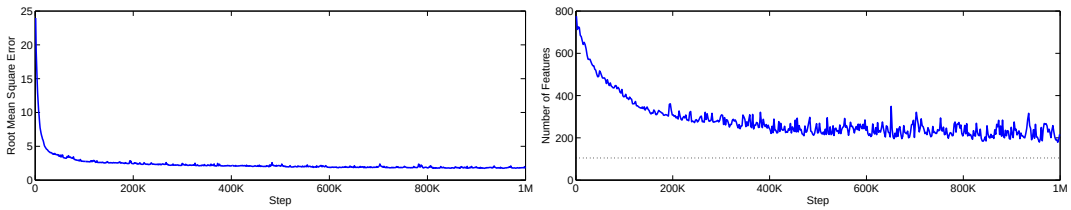
4.1 Blackjack

We tested a version of the bottomless-deck blackjack problem from Sutton and Barto [19], evaluating the policy in which the player stands on 20 or 21 and hits on anything else. In this problem, there are 200 states arising from the cross product of three variables: player total ([11...21]), dealer showing card value ([A...10]), and the presence of a usable ace in the player's hand (true/false). In our version, there are three additional states for win/lose/draw, and the player continues playing infinitely; after reaching a win/lose/draw state, the next state is drawn from the distribution of possible starting deals. With this modification, the problem is mixing with a well-defined stationary distribution. The player receives a reward of $+1$ for a win and -1 for a loss; all other rewards are zero. A discount factor of 0.95 keeps expected player losses bounded.

We provide the learner with 219 features: a bias term, an indicator for each of $\{\text{win}, \text{lose}\}$, indicator functions for all possible contiguous ranges of the player's current total times an indicator for whether the player has a usable ace, and likewise indicator functions on the dealer's show card times the ace indicator. Despite having more features than states, the rank of the feature matrix (the dimension of the basis) is only 41.

The fixed point for this problem was found via LARS-TD using full model information. Figure 1(a) shows the progress of the online algorithms vs. the fixed point when the regularization coefficient is set to 0.001. The step size for L1TD was set to 0.01, while the "fixed" step size (η) and the regular step size (α) were both set to 0.1 for L1TDAlt. Each algorithm was run for 3 million iterations in one continuous trajectory. The online algorithms quickly reach solutions which are near the fixed point, and make gradual progress thereafter.

Figure 1(b) shows how the sparsity of the solution changes over time. The plot shows a curve of the sorted absolute weight values after 100,000, 250,000, 1 million, and 3 million iterations of L1TD, as well as the weight curve for the fixed point solution (which has 16 non-zero weights). The figure has been scaled to emphasize the region of interest; the actual largest magnitude weights are around 28. The figure shows that, while L1TD produces few actual zeroes in this instance, over time it concentrates more and more weight into fewer and fewer features; most of the remaining non-zero features (which arise inevitably due to the stochastic updates) can be truncated without appreciably affecting the approximation.



(a) (b)

Figure 2: Mountain Car: (a) Root mean square error between L1TD and LARS-TD average value functions; (b) Number of features comprising 99% of total feature weight vs. number of steps.

Figure 1(c) shows the weights assigned to the top 18 features by L1TD, L1TDAlt, and LARS-TD; the strong agreement across the most significant features demonstrates the convergence of the online solutions to the fixed point qualitatively (in terms of the features it selects) as well as quantitatively (in terms of the learned value function).

4.2 Mountain Car

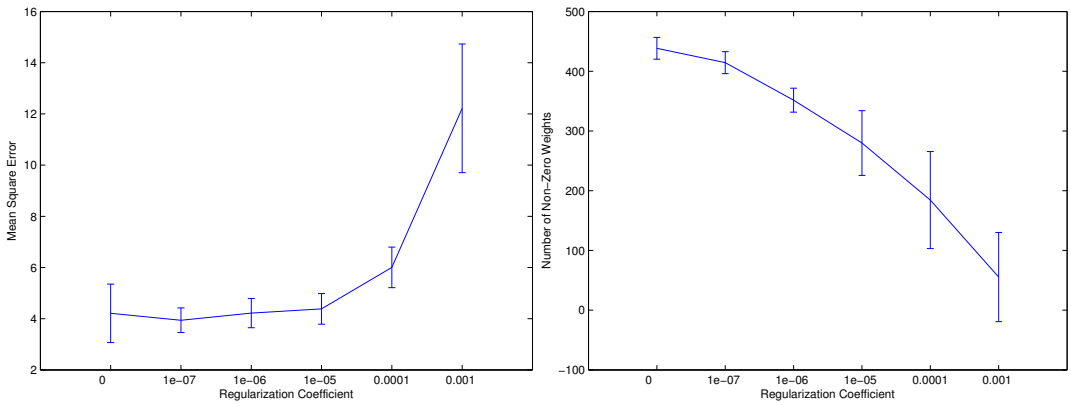
We also tested the familiar mountain car domain, in which an underpowered car must back up a small hill to gain enough potential energy to climb a larger one. The agent receives a -1 penalty for every step it takes until it reaches the goal region at the top of the hill. We chose to evaluate this task using a simple policy in which the car always accelerates in whichever direction it is already moving, or forward if it is at rest; while suboptimal, this policy ensures that the car will reach the goal region from any valid starting state. We used a discount factor of 0.99.

Our features for this task were composed of 1365 radial basis functions of various widths arranged in differently spaced grids over the state space together with a constant (bias) term. As movement in the mountain car domain is deterministic, we sampled trajectories by first selecting a starting state uniformly at random, and then following policy to the goal (on average, about 50 steps). A "ground truth" estimate of the fixed point was obtained by taking the average of the approximation weights returned by 100 runs of LARS-TD using independently collected samples from 2000 trajectories (approximately 100,000 samples for each run); the regularization coefficient was 0.01. A separate testing set of 2000 sample trajectories was collected and used in comparing the online algorithm (L1TD) with the ground truth estimate.

For online learning, we again start with a state selected uniformly at random, and follow policy to the goal; upon reaching the goal, we select a new state at random, and follow policy to the goal, etc. The algorithm was run for 1 million iterations using a fixed step size of 0.001. Figure 2(a) shows the root mean square error of the online value estimates for our test set versus the ground truth fixed point estimate; as with blackjack, the online algorithm quickly moves near the fixed point estimate, and then makes gradual improvements thereafter. Figure 2(b) shows generally how the sparsity of the online solution compares to the sparsity attained by LARS-TD; the plot gives the count of the largest features which comprise 99% or more of the total absolute weight value vs. the number of steps taken. The dotted black line in this figure shows the average number of features retained by LARS-TD in the ground truth estimation runs.

4.3 Pendulum

Finally, we tested the inverted pendulum domain, in which the task is to balance an inverted pendulum by applying forces to the cart to which it is attached. There are three (noisy) actions, applying force to the left or



(a) (b)

Figure 3: Pendulum: (a) Average mean square error between L1TDAlt and ground truth; (b) Average number of non-zero features.

right or no force. The agent receives a -1 penalty if and when the pendulum falls over. We evaluate this task using a simple, suboptimal policy that nevertheless balances the pendulum fairly well. We used a discount factor of 0.95.

For this task we construct a feature set by first transforming the state vector via PCA transform and a scaling matrix such that the resulting vectors live mostly in a 2π radius box around the origin. We then construct a wavelet-inspired basis using simple harmonic functions as the seed. The total number of features is 511. A set of 10,000 test samples was obtained from multiple on-policy trajectories in which the initial state was a vector randomly perturbed around the point at which the pendulum is balanced and unmoving, following the policy thereafter until the pendulum fell over. A ground truth estimate of the true value function at each sample was obtained by Monte Carlo sampling.

Our goal with this experiment was to demonstrate the performance of the algorithm for different values of regularization parameter as well as the sparsity induced by regularization. For each value of the regularization parameter, L1TDAlt was run ten times, each time for 5 million steps, starting with the state vector at the origin, thereafter following policy, resetting to the origin when the pendulum falls over. The step size (α) used was 0.02 while the shrinkage coefficient (η) was 0.5. Figure 3(a) shows the mean squared error of the online value estimates for our test set versus the ground truth estimate for various settings of the regularization parameter. As expected, small values of the regularization parameter lead to small errors. Note that the best result is obtained when the regularization is set to 10^{-7}, where the error and variance are lower than for the result with no regularization, showing a modest benefit in average performance and substantial reduction of variance. Figure 3(b) shows how the sparsity of the online solution varies with the regularization parameter.

5 Related Work

A number of authors have worked on approaches to regularization in reinforcement learning. In addition to LARS-TD [13], the L_1 regularized linear fixed point has been pursued by Johns et al. [12], who formulate the fixed point as a linear complementarity problem (LCP). Loth et al. [15] investigate Bellman residual minimization with L_1 regularization. L_1 regularization also appears in work by Petrik et al. [17], who give a regularized approximate linear programming solution for the more general reinforcement learning problem.

6 Conclusion

In this paper we have presented a novel regularized, online stochastic approximation algorithm, based on an iterative update equation whose fixed point corresponds to the L_1 regularized linear fixed point also shared by LARS-TD and LC-TD. We validated the ability of the online algorithm to converge to the fixed point described by theory in two sample problems, and demonstrated the tradeoff between sparsity and accuracy in a third problem.

More generally, we have established a connection between iterative and online algorithms for convex approximation with regularization and temporal difference learning with regularization. The soft-threshold shrinkage method extends naturally to L_2 and other forms of regularization, which can be applied to TD and analyzed following the approach taken above. Thus we hope to provide a template for future work in regularization algorithms for online TD learning.

Acknowledgments

This work was supported by NSF IIS-0713435. Opinions, findings, conclusions or recommendations herein are those of the authors and not necessarily those of NSF.

References

[1] Dimitri P. Bertsekas and John N. Tsitsiklis. *Neuro-Dynamic Programming*. Belmont, Mass: Athena Scientific, 1996.

[2] Vivek S. Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*. New York: Cambridge University Press, 2008.

[3] V.S. Borkar and S.P. Meyn. The ODE method for convergence of stochastic approximation and reinforcement learning. *SIAM Journal on Control and Optimization*, 38(2):447–469, 2000.

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

[5] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Review*, 43(1):129–159, March 2001.

[6] Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11): 1413–1457, 2004.

[7] David L. Donoho and Jain M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81 (3):425–455, 1994.

[8] John Duchi and Yoram Singer. Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research*, 10:2899–2934, Dec 2009.

[9] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–451, 2004.

[10] Mohammad Ghavamzadeh, Alessandro Lazaric, Rémi Munos, and Peter Auer. Finite-sample analysis of Lasso-TD. In *Proceedings of the Twenty-Eighth International Conference on Machine Learning (ICML-2011)*, 2011.

[11] Elaine T. Hale, Wotao Yin, and Yin Zhang. A fixed-point continuation method for ℓ_1-regularized minimization with applications to compressed sensing. Technical Report CAAM TR07-07, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, July 2007.

[12] Jeffrey Johns, Christopher Painter-Wakefield, and Ronald Parr. Linear complementarity for regularized policy evaluation and improvement. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1009–1017. 2010.

[13] J. Zico Kolter and Andrew Y. Ng. Regularization and feature selection in least-squares temporal difference learning. In Andrea Pohoreckyj Danyluk, Léon Bottou, and Michael L. Littman, editors, *ICML*, volume 382 of *ACM International Conference Proceeding Series*, page 66. ACM, 2009.

[14] John Langford, Lihong Li, and Tong Zhang. Sparse online learning via truncated gradient. In Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Léon Bottou, editors, *NIPS*, pages 905–912. MIT Press, 2008.

[15] Manuel Loth, Manuel Davy, and Philippe Preux. Sparse temporal difference learning using LASSO. In *ADPRL 2007*, 2007.

[16] M. R. Osborne, B. Presnell, and B. A. Turlach. A new approach to variable selection in least-squares problems. *IMA Journal of Numerical Analysis*, 20(3):389–403, 2000.

[17] Marek Petrik, Gavin Taylor, Ronald Parr, and Shlomo Zilberstein. Feature selection using regularization in approximate linear programs for markov decision processes. In Johannes Fürnkranz and Thorsten Joachims, editors, *ICML*, pages 871–878. Omnipress, 2010.

[18] Richard S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3: 9–44, 1988.

[19] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Cambridge, Mass.: MIT Press, 1998.

[20] Robert Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.