Sparse Temporal Difference Learning via Alternating Directions Method of Multipliers

Nikos Tsipinakis and James D. B. Nelson
Department of Statistical Science, University College London, UK
{nikolaos.tsipinakis.14, j.nelson}@ucl.ac.uk

***Abstract*—Recent work in off-line Reinforcement Learning has focused on efficient algorithms to incorporate feature selection, via ℓ_1-regularisation, into the Bellman operator fixed-point estimators. These developments now mean that over-fitting can be avoided when the number of samples is small compared to the number of features. However, it remains unclear whether existing algorithms have the ability to offer good approximations for the task of policy evaluation and improvement. In this paper, we propose a new algorithm for approximating the fixed-point based on the Alternating Direction Method of Multipliers (ADMM). We demonstrate, with experimental results, that the proposed algorithm is more stable for policy iteration compared to prior work. Furthermore, we also derive a theoretical result that states the proposed algorithm obtains a solution which satisfies the optimality conditions for the fixed point problem.**

I. Introduction

A core problem in off-line reinforcement learning (RL) emerges in situations where the state space is large and the dynamics are unknown. In such cases, explicit computation of the value functions becomes infeasible. Instead approximation techniques provide the only way forward. In particular, common choices to represent the value functions are those of linear architecture [24] where the hypothesis space $\mathcal{F}$ is defined by a set of feature vectors. In this domain, Least-Squares Temporal Difference (LSTD) algorithms [1], [8], [9] attempt to find the fixed point of the projected Bellman operator, ΠT, by using a rich number of samples. Unfortunately, in off-line learning, it is typically the case that the amount of available data is not sufficient, leading LSTD to poor predictions. Indeed, in the regression setting, when only a small number of samples are available relative to the number of features, the least-squares method is known to be very vulnerable to over-fitting. A typical way to overcome this issue is by incorporating ℓ_1- or ℓ_2-regularization known as *LASSO* [10] and *ridge*-regression [11], respectively. The former turns out to be of particular interest in the context of high-dimensional problems since it produces sparse solutions and therefore performs feature selection.

Many authors have explored regularized approximations for the value functions [3], [5], [12], [13] as a means to address the over-fitting problem in RL. However, none of these methods are able to both produce sparse solutions and treat the function approximation as a fixed-point problem. On the other hand, several recent methods has been proposed which add an ℓ_1 penalty to the fixed-point [2], [14]. Kolter and Ng [14] were the first to introduce the ℓ_1-regularization of the least-squares fixed-point. As the name suggests, their LARS-TD algorithm is inspired by the Least Angle Regression (LARS) algorithm. However, as is shown, the algorithm only converges to the fixed-point under some strong assumptions which rarely hold in the context of policy iteration. In [2] Johns et al. compute the same fixed point using the linear complementarity formulation but again the algorithm shares the same conditions with LARS-TD. Thus, there still remains an apparent need to introduce new algorithms to TD learning in order to efficiently evaluate the ℓ_1-regularized fixed-point problem within policy iteration.

In this work we propose to solve the ℓ_1-regularized fixed-point problem with the help of the ADMM [15]— a general optimisation framework which has recently received a wave of attention for various large regression problems. We also establish some theoretical properties of ADMM in the TD context. As in [2], [14], our problem formulation does not correspond to any convex optimization problem. However, our experimental results show that the proposed algorithm is able to find the fixed-point solution in both the prediction and control problem. More importantly, our results indicate a more efficient performance in the off-policy case compared to LARS-TD. On the other hand, LARS-TD is able to offer a richer set of solutions as a path method.

This paper is organized as follows. In Section II, we review the basic RL theory along with LSTD and LARS-TD. In Section III, we present the new ADMM algorithm for TD learning. In Section IV, we validate the efficiency of the algorithm through several experiment. In Section V, we discuss our contributions in the context of the most relevant, recent work and, finally, in Section VI we provide a general discussion for the proposed method and conclusions.

II. Preliminaries and Notation

A Markovian Decision Process (MDP) [16] is defined as the tuple $\langle S, A, P, R, \gamma \rangle$, where S denotes the set of states and $A(s)$ the set of actions available at each state; a policy, $\pi : S \to A$, is a mapping from states to actions; $P : S \times A \times S \to p(s'|s, \pi(s)) \in [0, 1]$ are the transition probabilities of moving to a new state, s', after executing an action $\pi(s) \in A(s)$. When reaching a new state the system returns a reward (or a cost), $R(s, \pi(s), s') : S \to \mathbb{R}$; $\gamma \in [0, 1)$ is the discount factor.

Nikos Tsipinakis is funded by a Dstl/UCL Impact studentship
James Nelson is partially supported by grants from the Dstl and Innovate UK/EPSRC

The quality of a policy can be determined by the value function defined as the expected discounted reward, $V^\pi(s) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t R(s_t)]$. For simplicity, we assume large finite state and action space, and thus the value function can be expressed in matrix form as a set of linear equations

$$V^\pi = R + \gamma P^\pi V^\pi.$$

If R and P are known, one can analytically solve the above linear system: $V^\pi = (I - \gamma P^\pi)^{-1} R$. Finally, the value function can be seen as the unique fixed point of the Bellman operator, $T : \mathbb{R}^{|S|} \to \mathbb{R}^{|S|}$, namely

$$V^\pi = T^\pi V^\pi, \quad T^\pi V^\pi = R + \gamma P^\pi V^\pi.$$

A. Function Approximation

We often deal with large action and state spaces where P and R are not known at hand, and hence deriving the value function explicitly is infeasible. In such situations, approximation of the value function is necessary. The most common approach is to employ a linear representation of the value function

$$\hat{V}^\pi = \Phi w = \sum_{i=1}^{|S|} \phi_i(s) w_i, \tag{1}$$

where $\Phi \in \mathbb{R}^{|S| \times n}$ is the feature matrix, and w the vector of the weights, $w \in \mathbb{R}^n$.

B. Least-Squares Temporal Difference

Approximating value function, as in (1), defines a hypothesis space spanned by the columns of Φ, $\mathcal{F} = \{\Phi w, w \in \mathbb{R}^n\}$. However, when applying the Bellman operator, the point $T^\pi \hat{V}^\pi$ does not necessarily lie on $\mathcal{F}$. LSTD [1] solves the problem of approximating the vector w by projecting $T^\pi \hat{V}^\pi$ back onto the hypothesis space. The objective function subject to approximation is therefore defined as, $\hat{V}^\pi = \Pi T^\pi \hat{V}^\pi$, where Π is the projection operator. As a consequence, LSTD searches for the fixed-point, $\hat{V}^\pi$, of the composed operator ΠT^π. The latter fixed-point problem can be then written as the optimization problem:

$$w = \underset{\theta \in \mathbb{R}^n}{\operatorname{argmin}} \|\Phi\theta - (R + \gamma P \Phi w)\|_\xi^2, \tag{2}$$

where $\xi \in \mathbb{R}^{|S| \times |S|}$ is a diagonal matrix with entries representing the stationary distribution of the states.

The fixed-point problem (2) requires knowledge of P and the construction of a large matrix, Φ. To this end LSTD collects m samples of the form, $\{s_i, a_i, r_i s_i'\}_{i=1,\dots,m}$, possibly sampled over several trajectories. This results in the sampled matrices

$$\tilde{\Phi} = \begin{pmatrix} \phi(s_1)^T \\ \vdots \\ \phi(s_m)^T \end{pmatrix}, \quad \tilde{\Phi}' = \begin{pmatrix} \phi(s_1')^T \\ \vdots \\ \phi(s_m')^T \end{pmatrix} \tilde{R} = \begin{pmatrix} r_1 \\ \vdots \\ r_m \end{pmatrix},$$

where $\Phi' = P\Phi$. Replacing the sampled features and reward matrices in (2) we have that

$$w = \underset{\theta \in \mathbb{R}^n}{\operatorname{argmin}} \|\tilde{\Phi}\theta - (\tilde{R} + \gamma \tilde{\Phi}' w)\|^2.$$

The above problem can be solved explicitly yielding the following set of linear equations

$$\tilde{A} w = \tilde{b},$$

where $\tilde{A} \in \mathbb{R}^{n \times n}$, $\tilde{b} \in \mathbb{R}^n$ are defined as $\tilde{\Phi}^T \left(\tilde{\Phi} - \gamma \tilde{\Phi}'\right)$ and $\tilde{\Phi}^T \tilde{R}$, respectively.

The main drawbacks of applying least-squares to function approximation emerge often if $m < n$. In this case, least-squares is prone to over-fitting and results in poor approximations, and also the matrix $\tilde{A}$ need not be full column rank— hence its left-inverse is not even guaranteed to exist.

C. LARS-TD

To overcome the limitations arising in LSTD, Kolter and Ng proposed LARS-TD [14]. They apply the ℓ_1-regularization penalty to the LSTD objective function which has the characteristic of avoiding over-fitting and performing feature selection. More precisely, the penalty is added to the projection of $T^\pi V$ onto the hypothesis space $\mathcal{F}$, which yields the following optimization problem

$$w = \underset{\theta \in \mathbb{R}^n}{\operatorname{argmin}} \|\tilde{\Phi}\theta - (\tilde{R} + \gamma \tilde{\Phi}' w)\|^2 + \lambda \|\theta\|_1. \tag{3}$$

The above optimization problem can be alternatively written as a fixed-point

$$\tilde{\Phi} w = \tilde{\Pi}_{\ell_1} \tilde{T}(\tilde{\Phi} w),$$

for which it has been proved that the operator $\tilde{\Pi}_{\ell_1} \tilde{T}$ is a γ-contraction, which in turn ensures the existence and uniqueness of the fixed-point $\tilde{\Phi} w$ [17]— although we note that w, itself, need not be unique.

Least Angle Regression (LARS) [18] is then adapted appropriately to problem (3) to efficiently solve for the ℓ_1-regularized fixed-point. Furthermore, LARS is based on a homotopy method which allows the computation of the complete regularization path. During this procedure, the algorithm maintains an active set, I, indicating the number of non zero elements of w. At each step, the regularization path is shrunken and a new element I is added or removed to $\tilde{A}_{I,I}$ and w_I in order for the optimality conditions never to be violated. It has been shown that as long as $\tilde{A}$ is a P-matrix[1], each LARS-TD step satisfies the optimality condition, and thus the algorithm always finds a solution to (3).

It is instructive to analytically review the major steps of the optimality conditions since the play a important role for LARS-TD and our proposed algorithm (cf. Section III). Define $G(\theta) = \frac{1}{2}\|\tilde{\Phi}\theta - (\tilde{R} + \gamma \tilde{\Phi}' w)\|^2 + \lambda \|\theta\|_1$, and thus the optimality condition for the convex problem is

$$0 \in \partial G(\theta), \tag{4}$$

[1] A square matrix A, not necessarily symmetric, is a P-matrix when all its principle minors are positive.

where ∂ denotes the sub-differential (since $\|\theta\|_1$ is not differentiable). Moreover, $\partial G(\theta) = \tilde{\Phi}^T(\tilde{\Phi}\theta - (\tilde{R} + \gamma\tilde{\Phi}'w)) + \lambda\partial\|\theta\|_1$, where:

$$\partial\|\theta\|_1 \in \begin{cases} \{+1\}, & \theta_i > 0 \\ [-1, 1], & \theta_i = 0 \\ \{-1\}, & \theta_i < 0, \end{cases}$$

Therefore, equation (4) implies that

$$[\tilde{\Phi}^T(\tilde{\Phi}\theta - (\tilde{R} + \gamma\tilde{\Phi}'w)]_i \in \begin{cases} \{-\lambda\}, & \theta_i > 0 \\ [-\lambda, \lambda], & \theta_i = 0 \\ \{\lambda\}, & \theta_i < 0. \end{cases}$$

Now, setting $w = \theta$, as required at the fixed point, the optimality conditions for the problem (3) become

$$[\tilde{b} - \tilde{A}w]_i \in \begin{cases} \{\lambda\}, & w_i > 0 \\ [-\lambda, \lambda], & w_i = 0 \\ \{-\lambda\}, & w_i < 0, \end{cases} \tag{5}$$

A solution w satisfying the optimality conditions yields the fixed point $\tilde{\Phi}w$ of the composed operator $\Pi_{\ell_1}T^\pi$.

LARS-TD enjoys many of the benefits of LARS, in that it follows a homotopy path and hence it offers all the solutions $w^*(\lambda)$. Its computational complexity is $\mathcal{O}(mnk^2)$, where k denotes the cardinality of the active set. Therefore, if the solution is sparse enough, the algorithm can compute the fixed-point very efficiently, i.e., after a few numbers of iterations. On the other hand, LARS-TD also inherits the LARS drawbacks, too. If the whole path need to be computed, the complexity reduces to a full least-squares, requiring to invert a nearly dense $\tilde{A}$, many times. Additionally, LARS-TD converges to the fixed point under the assumption that $\tilde{A}$ is a P-matrix. However, $\tilde{A}$ is not necessarily a P-matrix when samples are collected off-policy where, inevitably, the distribution of states is different from the distribution of the underlying policy. To overcome this issue, the authors propose adding an ℓ_2-penalty to the fixed-point problem, known as elastic net [19], to ensure that $\tilde{A}$ is positive definite. Elastic net formulation of the problem (3) nevertheless comes at cost of reduced sparsity. More importantly, in context of policy iteration, computing an almost complete regularization path could be inefficient, as also discussed in [2].

III. ADMM-TD

Our proposed approach is to apply ADMM to TD learning for solving the fixed-point problem (3). ADMM exploits some nice characteristics of the structure of (3) which matches those of ℓ_1-regularization in linear regression [15]. For this reason, we name the algorithm ADMM-TD.

A. The Algorithm

We proceed by deriving the ADMM steps. Problem (3) can be seen as an optimization problem of the form

$$\text{minimize} \quad \frac{1}{2}\|\tilde{\Phi}\theta - (R + \gamma\tilde{\Phi}'w)\| + \lambda\|\theta\|_1.$$

The above problem has the property of being separable and, hence, it can been split into two parts, namely $f(x)$ and $g(z)$. Furthermore, the requirement that the separate variables are equal yields the following equivalent problem:

$$\begin{aligned} &\text{minimize} \quad f(x) + g(z) \\ &\text{subject to} \quad x = z, \end{aligned}$$

where $f(x) = \frac{1}{2}\|\tilde{\Phi}\theta - (R + \gamma\tilde{\Phi}'w)\|$ and $g(z) = \lambda\|z\|_1$. The ADMM steps for the fixed point problem can be derived through the augmented Lagrangian which, in terms of the proximal form [20, Section 4.4], reduces to

$$\theta^{k+1} := \mathbf{prox}_{\mu f}(z^k - u^k) \tag{6}$$

$$z^{k+1} := \mathbf{prox}_{\mu g}(\theta^{k+1} + u^k) \tag{7}$$

$$u^{k+1} := u^k + \theta^{k+1} - z^{k+1}, \tag{8}$$

where $u = \frac{1}{\rho}y$ denotes the dual variable and $\mu = \frac{1}{\rho}$ the step-size parameter. The proximal operator of the first subproblem (6) is equal to

$$\theta^{k+1} := \underset{\theta}{\operatorname{argmin}} \left\{ \frac{1}{2}\|\tilde{\Phi}\theta - (R + \gamma\tilde{\Phi}'w)\| + \frac{\rho}{2}\|\theta - z^k + u^k\| \right\},$$

which can be solved by setting the gradient, w.r.t. θ, of $\frac{1}{2}\|\tilde{\Phi}\theta - (\tilde{R} + \gamma\tilde{\Phi}'w)\| + \frac{\rho}{2}\|\theta - z^k + u^k\|$. equal to zero:

$$\tilde{\Phi}^T(\tilde{\Phi}\theta - (\tilde{R} + \gamma\tilde{\Phi}'w)) + \rho(\theta - z^k + u^k) = 0$$

At the fixed point we require $w = \theta$, and thus it follows the fixed point solution

$$w^{k+1} := \left(\tilde{A} + \rho I\right)^{-1}\left(\tilde{b} + \rho(z^k - u^k)\right), \tag{9}$$

where $\tilde{A} = \tilde{\Phi}^T\left(\tilde{\Phi} - \gamma\tilde{\Phi}'\right)$ and $\tilde{b} = \tilde{\Phi}^T\tilde{R}$. Note that ρ equal to zero yields exactly the LSTD fixed point solution. For solving the second subproblem (7), we evaluate the proximal operator with respect to the previous iteration, i.e.,

$$z^{k+1} := \underset{z}{\operatorname{argmin}} \left\{ \|z\|_1 + \frac{\rho}{2}\|w^{k+1} - z + u^k\| \right\},$$

which reduces to the *soft-thresholding shrinkage operator* [21]

$$z^{k+1} := S_{\lambda/\rho}(w^{k+1} + u^k), \tag{10}$$

with

$$S_\beta(x) = \operatorname{sgn}(x) \odot \max\{|x| - \beta, 0\}, \tag{11}$$

and where $\operatorname{sgn}(x)$ is the signum function define as

$$\operatorname{sgn}(x) = \begin{cases} \{+1\}, & x_i > 0 \\ \{0\}, & x_i = 0 \\ \{-1\}, & x_i < 0. \end{cases}$$

The soft-thresholding is a component-wise operation, and thus $\odot$ denotes the component-wise multiplication.

The stopping criteria used in the following algorithm are the same with those discussed by the authors in [15, Section 3.3]. Note that the value w will be equal to z, and hence sparse, only in the limit. The ADMM-TD pseudocode is presented in Algorithm 1.

Algorithm 1 ADMM-TD

Input:
$\{s_i, r_i, s'_i\}_{i=1,\ldots,n}$ and form $\tilde{\Phi}$ and $\tilde{\Phi}'$
Initialize:
$\gamma \in [0,1]$, $\lambda \geq 0$, $\rho > 0$
$\tilde{A} \leftarrow \tilde{\Phi}^T \left(\tilde{\Phi} - \gamma \tilde{\Phi}'\right)$, $\tilde{b} \leftarrow \tilde{\Phi}^T R$
for $k = 0, 1, \ldots$ **do**
  $w^{k+1} \leftarrow (\tilde{A} + \rho I)^{-1}(\tilde{b} + \rho(z^k - u^k))$
  $z^{k+1} \leftarrow S_{\lambda/\rho}(w^{k+1} + u^k)$
  $u^{k+1} \leftarrow u^k + w^{k+1} - z^{k+1}$
end for
return w

B. Properties of ADMM-TD

In what follows, we show that the ADMM-TD fixed point solution, w, is also a solution to the ℓ_1-regularized fixed point problem (3) with optimality conditions (5).

Lemma 1. *The fixed point solution, w^*, as obtained from the ADMM-TD iterations in Algorithm 1, satisfies the optimality conditions (5), and is thus a solution to problem (3), for any $\lambda \geq 0$ and $\rho > 0$.*

Proof. At the fixed point, ADMM-TD iterations satisfy the following equations

$$w^* = (\tilde{A} + \rho I)^{-1}(\tilde{b} + \rho(z^* - u^*)) \tag{12}$$
$$z^* = S_{\lambda/\rho}(w^* + u^*) \tag{13}$$
$$u^* = u^* + w^* - z^*. \tag{14}$$

Equation (14) implies that $w^* = z^*$. From (12) it follows that

$$u^* = \frac{1}{\rho}(\tilde{b} - \tilde{A}w^*). \tag{15}$$

Similarly, equation (10) can be rewritten as:

$$w^* = S_{\lambda/\rho}(w^* + u^*). \tag{16}$$

Now, combining (15) with (16) we have that

$$w^* = S_{\lambda/\rho}\left(w^* + \frac{1}{\rho}(\tilde{b} - \tilde{A}w^*)\right). \tag{17}$$

From this point, the proof parallels those in [21] and [22]. Using now the definition of the shrinkage operator (11), the right-hand side of the equation (17) can be written as

$$\operatorname{sgn}\left(w_i^* + \frac{1}{\rho}[\tilde{b} - \tilde{A}w^*]_i\right) \max\left\{\left|w_i^* + \frac{1}{\rho}[\tilde{b} - \tilde{A}w^*]_i\right| - \frac{\lambda}{\rho}, 0\right\}.$$

Since the $\max$ operator is nonnegative, the sign of operator sgn must agree with the sign of w_i^*. Therefore, if $w_i^* > 0$, it follows that

$$\operatorname{sgn}\left(w_i^* + \frac{1}{\rho}[\tilde{b} - \tilde{A}w^*]_i\right) = 1,$$

and also

$$\max\left\{\left|w_i^* + \frac{1}{\rho}[\tilde{b} - \tilde{A}w^*]_i\right| - \frac{\lambda}{\rho}, 0\right\} = w_i + \frac{1}{\rho}[\tilde{b} - \tilde{A}w^*]_i - \frac{\lambda}{\rho}.$$

Replacing the above results to the equation (17) we have that

$$[\tilde{b} - \tilde{A}w^*]_i = \lambda, \quad w_i^* > 0,$$

as the optimality conditions (5) indicate. With similar operations one can show that $[\tilde{b} - \tilde{A}w^*]_i = -\lambda$, for any $w_i^* < 0$. Finally, $w_i^* = 0$ implies either that

$$\operatorname{sgn}\left(\frac{1}{\rho}[\tilde{b} - \tilde{A}w^*]_i\right) = 0, \tag{18}$$

or

$$\max\left\{\left|\frac{1}{\rho}[\tilde{b} - \tilde{A}w^*]_i\right| - \frac{\lambda}{\rho}, 0\right\} = 0. \tag{19}$$

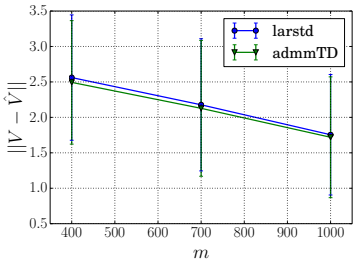
In the first case, (18), we must have that $[\tilde{b} - \tilde{A}w^*]_i = 0$, which satisfies the optimality conditions. From the second case, (19), it follows that $|\frac{1}{\rho}[\tilde{b} - \tilde{A}w^*]_i| \leq \frac{\lambda}{\rho} \Rightarrow -\lambda \leq [\tilde{b} - \tilde{A}w^*]_i \leq \lambda$, which concludes the proof. □

The above Lemma shows that the ADMM-TD solves the fixed point problem (3), and that the above proof also holds for $w = z$. A convergence proof of the ADMM-TD to the fixed point remains outstanding. However, our experimental results in Section IV below indicate comparable, if not better, behavior relative to the LARS-TD algorithm. In particular, unlike LARS-TD, the proposed ADMM-TD converges to the fixed point in both on- and off-sampling cases.

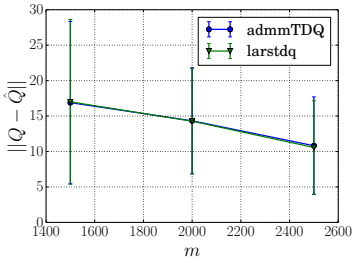
IV. Experiments

The four-rooms grid problem, as discussed in [23, Section 5], was used to compare the proposed ADMM-TD with LARS-TD. The problem involves a two dimensional grid with total number of states $S = M \times N$, where M and N are chosen as the largest factors of S. The grid is split into four interconnected rooms where only the neighbor rooms are connected to each other. Goal states are the states $S-1, S-2$. The agent receives a reward of 1 when it visits the goal states and receives -1 elsewhere. The action set available to the agent comprises eight actions (towards all possible directions) —agents with this characteristics are called "king-move" agents. Each action has a probability of success of 0.85. We use the four-rooms environment with a total of 25 states in all our experiments. The value functions are represented with Gaussian Radial Basis Functions (RBFs) concatenated over different two-dimensional grids. Training samples are collected in different episodes of 5 steps each. The step-size parameter is kept fixed for all the experiment, $\rho = 0.1$. To select the most effective value of the regularization parameter, we use K-fold cross-validation, for a total of 100 values of λ, with K either 5 or 10 according to the magnitude of samples.

In the first experiment, Figure 1a, we approximate the state-value function using 1365 features concatenated over $[2, 4, 8, 16, 32]$ grids. We compare the performance of the algorithms over different number of samples $(400, 700, 1000)$ collected on-policy, in 50 trials. As expected, LARS-TD and the LASSO formulation of ADMM-TD yielded similar averaged approximation errors.

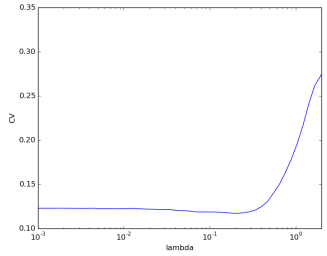


(a)

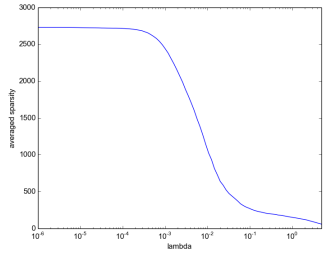


(b)

Fig. 1. (a) Averaged approximation error over 50 trials for V function using 1365 features versus samples m. (b) Averaged approximation error over 50 trials for Q function using 2728 features versus samples m.



(a)



(b)

Fig. 2. (a) 10-fold cross-validation (CV) versus regularization parameter λ; minimum CV achieved for $\lambda = 0.214$. (b) averaged sparsity versus regularization parameter λ; $\lambda = 0.214$ yields approximately 227 nonzero features.

Subsequently, we test both algorithms in the context of Q value approximation. In this experiment we supply both methods with 2728 features concatenated over $[2, 4, 8, 16]$ grids. Again, we average our results over 50 trials using different number of samples (see Figure 1b). This time, we collect our samples off-policy (executing a random policy each time).

TABLE I
MEAN SIMULATED REWARD (20 TRIALS) $\pm$ STANDARD ERROR BETWEEN ADMM-TD AND LARS-TD FOR $m = (1500, 2000)$ SAMPLES AND 2728 FEATURES.

No. of samples	averaged simulated reward	
	ADMM-TD	LARS-TD
1500	73.06 ± 5.001	66.76 ± 6.91
2000	73.77 ± 3.47	67.78 ± 7.14

Under these circumstances, LARS-TD was not always able to find a solution. In particular, we found LARS-TD to violate the optimality conditions $27/150$ times, while ADMM-TD never failed. For this reason, the LARS-TD results, illustrated in Figure 1b, incorporate ℓ_2-regularization as proposed in [14]. The results, though, indicate identical behavior for both algorithms, showing the same averaged approximation error, even with the elastic net formulation of LARS-TD. However, this modification in LARS-TD comes with the drawback of increased computational cost due to reduced sparsity. For instance, 10-fold cross-validation in the case of $m = 1500$ indicates $\lambda = 0.214$ producing about 200 nonzero features (Figures 2a, 2b), while for the same example, LARS-TD produces approximately 2000 nonzero features.

In the final experiment, the ability of both algorithms to find good policies (policy iteration) is evaluated. We use $(1500, 2000)$ samples and, as before, the samples are collected in the same manner. The results are averaged over 20 trials where each policy iteration trial is run until either convergence to the optimal solution or a maximum of 15 steps is reached. In this setting, we found that LARS-TD violated the optimality conditions repeatedly, and hence was never able to find a good policy. This is understandable because the policy changes drastically at each step due to the rich available action set. On the other hand, given enough samples, ADMM-TD never failed to reach the optimal policy —we only found ADMM-TD not satisfying the optimality conditions for $m < 1000$. Therefore, in order to compare both methods in terms of the simulated reward, we, again, aplly the an ℓ_2 penalty in LARS-TD algorithm. Nevertheless, LARS-TD now requires storing and inverting a square matrix with almost 2000 entries (as described in the previous paragraph) many times at each policy iteration step. The fact that LARS-TD computes a complete homotopy path within policy iteration makes the algorithm inefficient with respect to time complexity (the same issue is also discussed in [2]). As a result, for practical purposes, we tuned both algorithms to produce no more than 200 nonzero features. In this context, ADMM-TD yielded better policies compared to LARS-TD, as shown in Table I. Furthermore, we noted that approximately 5 steps were needed for ADMM-TD to reach an optimal policy, while LARS-TD needed always more than 10 steps.

V. RELATED WORK

There has been several works recently which perform feature selection in Temporal Difference learning. In [2], the ℓ_1-regularized fixed point problem is solved as Linear

Complementarity Problem rather than using LARS algorithm. This approach overcomes the limitations of LARS-TD in the context of policy iteration by allowing for warm-starts at each step. However, for finding a solution to (3), LCP requires $\tilde{A}$ to be a P-matrix. Loth et al. [3] perform ℓ_1-regularization to the Bellman Residual Minimization which cannot then be considered as a fixed point problem. Furthermore, Petrick et al. [4] propose ℓ_1-regularization in the context of approximate linear program which suffers when applied to noisy samples. Geist and Scherrer [5] take a different route by solving a convex optimization problem. They apply ℓ_1-regularization to the Projected Bellman Residual (PBR), instead. Although this formulation enjoys the benefits of a convex problem, it comes with the increased computational cost requiring the calculation of the projection, $\tilde{\Pi}$, and moreover it cannot then be interpreted as a fixed-point problem.

Finally, the only other ADMM approach to RL that we are aware of, called ADMM-BPDN [6], solves the ℓ_1-PBR problem as a constrained convex problem, with $\|\tilde{C}w+\tilde{d}\| \leq \epsilon$. This formulation reduces to a first-order method which effectively corresponds to the *basis pursuit denoising problem* (BPDN) [7]. It has been proved that the algorithm converges to a solution if $\tau \leq \frac{1}{\text{eig}_{\max}(\tilde{C}^T\tilde{C})}$, where τ is the proximal step-size parameter. However, $\text{eig}_{\max}(\tilde{C}^T\tilde{C})$ is often too large and hence it is unclear whether the method is efficient, in terms of time complexity, when dealing with large problems.

VI. Conclusion

We here proposed an alternative off-line algorithm for solving the ℓ_1-regularized fixed-point. We validated the efficacy of our algorithm against LARS-TD in a complex experimental environment with many available actions. Results indicate that, given enough samples, ADMM-TD is able to find the fixed-point solution even within the policy iteration procedure.

The advantage of ADMM-TD as a direct method committed to only a single value of λ is that, even in the case where the optimality conditions are violated, one may discard the coefficient for the specific λ without affecting the other solutions. However, when LARS-TD violates the optimality conditions, the complete homotopy path is affected. As also shown in [14], the homotopy path reaches discontinuities which makes the algorithm return multiple fixed-point. On the other hand, LARS-TD, as path algorithm, is able to return the complete set of solutions, while ADMM-TD only returns a subset over a fixed grid. Further, we demonstrated that LARS-TD modified to incorporate ℓ_2-regularization loses its computational efficiency due to the decreased sparsity. We also note that, similar to the usual ADMM, the proposed ADMM-TD can be easily extended to allow other forms of regularization (eg. Tikhonov regularization or elastic net).

As a future direction, we plan for a proof of convergence of the ADMM-TD. We conjecture that the algorithm could be shown to converge to the fixed point under much weaker assumption compared to the existing work. Our anticipation is driven, first, from the fact that the step-size parameter ρ is incorporated in the diagonal of $\tilde{A}$ and, second, due to the behavior of our algorithm in the context of policy iteration. We also plan to apply ADMM-TD to control problems.

Acknowledgments

The authors would like to thank A. J. Weinstein for kindly sharing his code with us.

References

[1] S. Bradtke and A. Barto, Linear least-squares algorithms for temporal difference. *Machine Learning*. 22, 33-57, 1996.

[2] J. Johns and C. Painter-Wakefield, Linear complementarity for regularized policy evaluation and emprovement. In *Advances in Neural Information Processing Systems*. 23, 1009-1017, 2010.

[3] M. Loth, M. Davy and P. Preux, Sparse temporal difference learning using LASSO. *IEEE, International Symposium on Approximate Dynamic Programming and Reinforcement Learning*. 352-359, 2007.

[4] M. Petrik, G. Taylor, R. Parr and S. Zilberstein, Feature selection using regularization in approximate linear programs for Markov decision processes. In *Proccedings of the 27th International Conference on Machine Learning*. 27, 871-878, 2010.

[5] M. Geist, and B. Scherrer, ℓ_1-Penalized projected bellman residual. *Recent Advances in Reinforcement Learning*. 89-101, 2012.

[6] Z. Qin, W. Li and F. Janoos, Sparse reinforcement learning via convex optimization. In *Proccedings of the 31st International Conference on Machine Learning*. 31, 424-432, 2014.

[7] J. Yang and Y. Zhang, Alternating direction algorithms for ℓ_1-problems in compressive sensing. *SIAM Journal on Scientific Computing*. 33, 250-278, 2011.

[8] J. Boyan, Technical update: least-squares temporal difference learning. *Machine Learning*. 49, 233-249, 2002.

[9] M. Lagoudakis and R. Parr, Least-squares policy iteration. *Journal of Machine Learning Research*. 4, 1107-1149, 2003.

[10] R. Tibshirani, Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistics Society B*. 58, 267-288, 1996.

[11] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001.

[12] A. M. Farahmand, M. Ghavamzadeh, C. Szepesvari and S. Mannor, Regularized policy iteration. In *Advances in Neural Information Processing Systems*. 21, 441-448, 2008.

[13] M. Geist, B. Scherrer, A. Lazaric and M. Ghavamzadeh, A Dantzig Selector Approach to Temporal Difference Learning. In *Proccedings of the 29th International Conference on Machine Learning*. 2012.

[14] Z. Kolter and A. Ng, Regularization and feature selection in least-squares temporal difference. In *Proccedings of the 26th International Conference on Machine Learning*. 521-528, 2009.

[15] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers. *Machine Learning*. 3(1), 1-123, 2010.

[16] R. Sutton and A. Barto, *An introduction to Reinforcement Learning*. MIT Press, 1998.

[17] M. Ghavamzadeh, A. Lazaric, R. Munos and M. Hoffman, Finite sample analysis of Lasso-TD. In *Proccedings of the 28th International Conference on Machine Learning*. 2011.

[18] B. Efron, T. Hastie, I. Johnstone and R. Tibshirani, Least angle regression. *Annals of Statistics*. 32, 407-499, 2004.

[19] H. Zou and T. Hastie, Regularization and variable selection via the elastic net. *Journal of the Royal Statistics Society B*. 67, 301-320, 2007.

[20] N. Parikh and S. Boyd, Proximal Algorithms. *Foundations and Trends in Optimization*. 1(3), 123-231, 2013.

[21] E. T. Hale, W. Yin and Y. Zhang, A fixed-point continuation method for ℓ_1-regularized minimization with applications to compressed sensing. Technical report: Department of Computational and Applied Mathematics, Rice University, Houston, Texas, CAAM TR07-07, 2007.

[22] C. Painter-Wakefield and R. Parr, $L1$ regularized linear temporal difference learning. Technical report: Department of Computer Science, Duke University, Durham, NC, TR-2012-01, 2012.

[23] A. J. Weinstein, Inference and Learning in High-Dimensional Spaces. PhD Thesis. Department of Electrical Engineering and Computer Science, Colorado School of Mines, Golden, Colorado, 2013.

[24] J. Tsitsiklis and B. V. Roy, An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*. 42, 674-690, 1997.