Consistent On-Line Off-Policy Evaluation

Assaf Hallak [1] **Shie Mannor** [1]

Abstract

The problem of on-line off-policy evaluation (OPE) has been actively studied in the last decade due to its importance both as a stand-alone problem and as a module in a policy improvement scheme. However, most Temporal Difference (TD) based solutions ignore the discrepancy between the stationary distribution of the behavior and target policies and its effect on the convergence limit when function approximation is applied. In this paper we propose the Consistent Off-Policy Temporal Difference (COP-TD(λ, β)) algorithm that addresses this issue and reduces this bias at some computational expense. We show that COP-TD(λ, β) can be designed to converge to the same value that would have been obtained by using on-policy TD(λ) with the target policy. Subsequently, the proposed scheme leads to a related and promising heuristic we call log-COP-TD(λ, β). Both algorithms have favorable empirical results to the current state of the art online OPE algorithms. Finally, our formulation sheds some new light on the recently proposed Emphatic TD learning.

1. Introduction

Reinforcement Learning (RL) techniques were successfully applied in fields such as robotics, games, marketing and more (Kober et al., 2013; Al-Rawi et al., 2015; Barrett et al., 2013). We consider the problem of off-policy evaluation (OPE) – assessing the performance of a complex strategy without applying it. An OPE formulation is often considered in domains with limited sampling capability. For example, marketing and recommender systems (Theocharous and Hallak, 2013; Theocharous et al., 2015) directly relate policies to revenue. A more extreme example is drug administration, as there are only few patients in the testing population, and sub-optimal policies can have life threatening effects (Hochberg et al., 2016). OPE can also be useful as a module for policy optimization in a policy improvement scheme (Thomas et al., 2015a).

In this paper, we consider the OPE problem in an on-line setup where each new sample is immediately used to update our current value estimate of some previously unseen policy. We propose and analyze a new algorithm called COP-TD(λ,β) for estimating the value of the target policy; COP-TD(λ,β) has the following properties:

1. Easy to understand and implement on-line.
2. Allows closing the gap to consistency such that the limit point is the same that would have been obtained by on-policy learning with the target policy.
3. Empirically comparable to state-of-the art algorithms.

Our algorithm resembles (Sutton et al., 2015)'s Emphatic TD that was extended by (Hallak et al., 2015) to the general parametric form ETD(λ,β). We clarify the connection between the algorithms and compare them empirically. Finally, we introduce an additional related heuristic called Log-COP-TD(λ,β) and motivate it.

2. Notations and Background

We consider the standard discounted Markov Decision Process (MDP) formulation (Bertsekas and Tsitsiklis, 1996) with a single long trajectory. Let $M = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \zeta, \gamma)$ be an MDP where $\mathcal{S}$ is the finite state space and $\mathcal{A}$ is the finite action space. The parameter $\mathcal{P}$ sets the transition probabilities $\Pr(s'|s,a)$ given the previous state $s \in \mathcal{S}$ and action $a \in \mathcal{A}$, where the first state is determined by the distribution ζ. The parameter $\mathcal{R}$ sets the reward distribution $r(s,a)$ obtained by taking action a in state s and γ is the discount factor specifying the exponential reduction in reward with time. The process advances as follows:

A state s_0 is sampled according to the distribution $\zeta(s)$. Then, at each time step t starting from $t = 0$ the agent draws an action a_t according to the stochastic behavior policy $\mu(a|s_t)$, a reward $r_t \doteq r(s_t, a_t)$ is accumulated by the agent, and the next state s_{t+1} is sampled using the transition probability $\Pr(s'|s_t, a_t)$.

[1] The Technion, Haifa, Israel. Correspondence to: Assaf Hallak <ifogph@gmail.com>, Shie Mannor <shie@ee.technion.ac.il>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017. Copyright 2017 by the author(s).

The expected discounted accumulated reward starting from a specific state and choosing an action by some policy π is called the value function, which is also known to satisfy the Bellman equation in a vector form:

$$V^\pi(s) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t r_t \,\middle|\, s_0 = s\right],\; T_\pi V \doteq R_\pi + \gamma P_\pi V,$$

where $[R_\pi]_s \doteq \mathbb{E}_\pi\left[r(s,\pi(s))\right]$ and $[P_\pi]_{s,s'} \doteq \mathbb{E}_\pi\left[\Pr(s'|s,\pi(s))\right]$ are the policy induced reward vector and transition probability matrix respectively; T_π is called the Bellman operator. The problem of estimating $V^\pi(s)$ from samples is called policy evaluation. If the target policy π is different than the behavior policy μ which generated the samples, the problem is called off-policy evaluation (OPE). The TD(λ) (Sutton, 1988) algorithm is a standard solution to on-line on-policy evaluation: Each time step the temporal difference error updates the current value function estimate, such that eventually the stochastic approximation process will converge to the true value function. The standard form of TD(λ) is given by:

$$\begin{aligned} R^{(n)}_{t,s_t} =& \sum_{i=0}^{n-1}\gamma^i r_{t+i} + \gamma^n \hat{V}_t(s_{t+n}), \\ R^\lambda_{t,s_t} =& (1-\lambda)\sum_{n=0}^{\infty}\lambda^n R^{(n+1)}_{s_t}, \\ \hat{V}_{t+1}(s_t) =& \hat{V}_t(s_t) + \alpha_t\left(R^\lambda_{t,s_t} - \hat{V}_t(s_t)\right), \end{aligned} \tag{1}$$

where α_t is the step size. The value $R^{(n)}_{t,s_t}$ is an estimate of the current state's $V(s_t)$, looking forward n steps, and R^λ_{t,s_t} is an exponentially weighted average of all of these estimates going forward till infinity. Notice that Equation 1 does not specify an on-line implementation since $R^{(n)}_{t,s_t}$ depends on future observations, however there exists a compact on-line implementation using eligibility traces (Bertsekas and Tsitsiklis (1996) for on-line TD(λ), and Sutton et al. (2014), Sutton et al. (2015) for off-policy TD(λ)). The underlying operator of TD(λ) is given by:

$$\begin{aligned} T^\lambda_\pi V &= (1-\lambda)\sum_{n=0}^{\infty}\lambda^n\left(\sum_{i=0}^{n}\gamma^i P^i_\pi R_\pi + \gamma^{n+1}P^{n+1}_\pi V\right) \\ &= (1-\lambda)(I-\lambda T_\pi)^{-1}T_\pi V, \end{aligned}$$

and is a $\frac{\gamma(1-\lambda)}{1-\lambda\gamma}$-contraction (Bertsekas, 2012).

We denote by $d_\mu(s)$ the stationary distribution over states induced by taking the policy μ and mark $D_\mu = diag(d_\mu)$. Since we are concerned with the behavior at infinite horizon, we assume $\zeta(s) = d_\mu(s)$. In addition, we assume that the MDP is ergodic for the two specified policies μ, π so $\forall s \in \mathcal{S} : d_\mu(s) > 0, d_\pi(s) > 0$ and that the OPE problem is proper – $\pi(a|s) > 0 \Rightarrow \mu(a|s) > 0$.

When the state space is too large to hold $V^\pi(s)$, a linear function approximation scheme is used: $V^\pi(s) \approx \theta_\pi^\top \phi(s)$, where θ is the optimized weight vector and $\phi(s)$ is the feature vector of state s composed of k features. We denote by Π_{d_π} the projection to the subspace spanned by the features with respect to the d_π-weighted norm, and by $\Phi \in \mathbb{R}^{S,k}$ the matrix whose lines consist of the feature vectors for each state and assume its columns are linearly independent.

TD(λ) can be adjusted to find the fixed point of $\Pi_{d_\pi}T^\lambda_\pi$ (Sutton and Barto, 1998):

$$\begin{aligned} R^{(n)}_{t,s_t} =& \sum_{i=0}^{n-1}\gamma^i r_{t+i} + \gamma^n \theta_t^\top \phi(s_{t+n}), \\ R^\lambda_{t,s_t} =& (1-\lambda)\sum_{n=0}^{\infty}\lambda^n R^{(n+1)}_{s_t}, \\ \theta_{t+1} =& \theta_t + \alpha_t\left(R^\lambda_{t,s_t} - \theta_t^\top\phi(s_t)\right)\phi(s_t). \end{aligned}$$

Finally, we define OPE-related quantities:

$$\rho_t \doteq \frac{\pi(a_t|s_t)}{\mu(a_t|s_t)}, \quad \Gamma^n_t \doteq \prod_{i=0}^{n-1}\rho_{t-1-i}, \quad \rho_d(s) \doteq \frac{d_\pi(s)}{d_\mu(s)},$$

we call ρ_d the covariate shift ratio (as denoted under different settings by (Hachiya et al., 2012)).

We summarize the assumptions used in the proofs:

1. For both policies the induced Markov chain is ergodic.
2. The first state s_0 is distributed according to the stationary distribution of the behavior policy $d_\mu(s)$.
3. The problem is proper: $\pi(a|s) > 0 \Rightarrow \mu(a|s) > 0$.
4. The feature matrix Φ has full rank k.

Assumption 1 is commonly used for convergence theorems as it verifies the value function is well defined on all states regardless of the initial sampled state. Assumption 2 can be relaxed since we are concerned with the long-term properties of the algorithm past its mixing time – we require it for clarity of the proofs. Assumption 3 is required so the importance sampling ratios will be well defined. Assumption 4 guarantees the optimal θ is unique which greatly simplifies the proofs.

3. Previous Work

We can roughly categorize previous OPE algorithms to two main families. Gradient based methods that perform stochastic gradient descent on error terms they want to minimize. These include GTD (Sutton et al., 2009a), GTD-2, TDC (Sutton et al., 2009b) and HTD (White and White, 2016). The main disadvantages of gradient based methods are (A) they usually update an additional error correcting term, which means another time-step parameter needs to be controlled; and (B) they rely on estimating non-trivial terms, an estimate that tends to converge slowly. The other family uses importance sampling (IS) methods that correct the gains between on-policy and off-policy updates using the IS-ratios ρ_t's. Among these are full IS (Precup et al., 2001) and ETD(λ,β) (Sutton et al., 2015). These methods are characterized by the bias-variance trade-off they resort to – navigating between biased convergent values (or even divergent), and very slow convergence stemming from the high variance of IS correcting factors (the ρ_t products). There are also a few algorithms that fall between the two, for example TO-GTD (van Hasselt et al., 2014) and WIS-TD(λ) (Mahmood and Sutton, 2015).

A comparison of these algorithms in terms of convergence rate, synergy with function approximation and more is available in (White and White, 2016; Geist and Scherrer, 2014). We focus in this paper on the limit point of the convergence. For most of the aforementioned algorithms, the process was shown to converge almost surely to the fixed point of the projected Bellman operator $\Pi_d T_\pi$ where d is some stationary distribution (usually d_μ), however the d in question was never[1] d_π as we would have obtained from running on-policy TD with the target policy (also see (Kolter, 2011) for relevant discussion). The algorithm achieving the closest result is ETD(λ,β) which replaced d with $f = \left(I - \beta P_\pi^\top\right)^{-1} d_\mu$, where β trades-off some of the process' variance with the bias in the limit point. Hence, our main contribution is a consistent algorithm which can converge to the same value that would have been obtained by running an on-policy scheme with the same policy.

4. Motivation

Here we provide a motivating example showing that even in simple cases with "close" behavior and target policies, the two induced stationary distributions can differ greatly. Choosing a specific linear parameterization further emphasizes the difference between applying on-policy TD with the target policy, and applying inconsistent off-policy TD.

Assume a chain MDP with numbered states $1, 2, ..|S|$, where from each state s you can either move left to state $s-1$, or right to state $s+1$. If you've reached the beginning or the end of the chain (states 1 or $|S|$) then taking a step further does not affect your location. Assume the behavior policy moves left with probability $0.5+\epsilon$, while the target policy moves right with probability $0.5+\epsilon$. It is easy to see that the stationary distributions are given by:

$$d_\mu(s) \propto \left(\frac{0.5-\epsilon}{0.5+\epsilon}\right)^s, \qquad d_\pi(s) \propto \left(\frac{0.5+\epsilon}{0.5-\epsilon}\right)^s.$$

For instance, if we have a length 100 chain with $\epsilon = 0.01$, for the rightmost state we have $d_\mu(|S|) \approx 8 \cdot 10^{-4}, d_\pi(|S|) \approx 0.04$. Let's set the reward to be 1 for the right half of the chain, so the target policy is better since it spends more time in the right half. The value of the target policy in the edges of the chain for $\gamma = 0.99$ is $V^\pi(1) = 0.21, V^\pi(100) = 99.97$.

Now what happens if we try to approximate the value function using one constant feature $\phi(s) \equiv 1$? The fixed point of $\Pi_{d_\mu} T_\pi$ is $\theta = 11.92$, while the fixed point of $\Pi_{d_\pi} T_\pi$ is $\theta = 88.08$ – a substantial difference. The reason for this difference lies in the emphasis each projection puts on the states: according to Π_{d_μ}, the important states are in the left half of the chain – these with low value function, and therefore the value estimation of all states is low. However, according to Π_{d_π} the important states are concentrated on the right part of the chain since the target policy will visit these more often. Hence, the estimation error is emphasized on the right part of the chain and the value estimation is higher. When we wish to estimate the value of the target policy, we want to know what will happen if we deploy it instead of the behavior policy, thus taking the fixed point of $\Pi_{d_\pi} T_\pi$ better represents the off-policy evaluation solution.

5. COP-TD(λ, β)

Most off-policy algorithms multiply the TD summand of TD(λ) with some value that depends on the history and the current state. For example, full IS-TD by (Precup et al., 2001) examines the ratio between the probabilities of the trajectory under both policies:

$$\frac{P_\pi(s_0, a_0, s_1, \dots, s_t, a_t)}{P_\mu(s_0, a_0, s_1, \dots, s_t, a_t)} = \prod_{m=0}^{t} \rho_m = \Gamma_t^t \rho_t. \quad (2)$$

In problems with a long horizon, or these that start from the stationary distribution, we suggest using the time-invariant covariate shift ρ_d multiplied by the current ρ_t. The intuition is the following: We would prefer using the probabilities ratio given in Equation 2, but it has very high variance, and after many time steps we might as well look at the stationary distribution ratio instead. This direction leads us to the following update equations:

$$\theta_{t+1} = \\ \theta_t + \alpha_t \rho_d(s_t) \rho_t \left(r_t + \theta_t^\top \left(\gamma\phi(s_{t+1}) - \phi(s_t)\right)\right) \phi(s_t). \quad (3)$$

Lemma 1. *If the α_t satisfy $\sum_{t=0}^\infty \alpha_t = \infty, \sum_{t=0}^\infty \alpha_t^2 < \infty$ then the process described by Eq. (3) converges almost surely to the fixed point of $\Pi_\pi T_\pi V = V$.*

[1] Except full IS, however its variance is too high to be applicable in practice.

The proof follows the ODE method (Kushner and Yin, 2003) similarly to Tsitsiklis and Van Roy (1997) (see the appendix for more details).

Since $\rho_d(s)$ is generally unknown, it is estimated using an additional stochastic approximation process. In order to do so, we note the following Lemma:

Lemma 2. *Let $\widehat{\rho_d}$ be an unbiased estimate of ρ_d, and for every $n = 0, 1, \ldots, t$ define $\tilde{\Gamma}_t^n \doteq \widehat{\rho_d}(s_{t-n})\Gamma_t^n$. Then:*

$$\mathbb{E}_\mu \left[\tilde{\Gamma}_t^n | s_t \right] = \rho_d(s_t).$$

For any state s_t there are $t \to \infty$ such quantities $\{\tilde{\Gamma}_t^n\}_{n=0}^t$, where we propose to weight them similarly to TD(λ):

$$\tilde{\Gamma}_t^\beta = (1-\beta) \sum_{n=0}^{\infty} \beta^n \tilde{\Gamma}_t^{n+1}.$$

Note that $\rho_d(s)$, unlike $V(s)$, is restricted to a close set since its d_μ-weighted linear combination is equal to 1 and all of its entries are non-negative; We denote this d_μ-weighted simplex by Δ_{d_μ}, and let $\Pi_{\Delta_{d_\mu}}$ be the (non-linear) projection to this set with respect to the Euclidean norm ($\Pi_{\Delta_{d_\mu}}$ can be calculated efficiently, (Chen and Ye, 2011)). Now, we can devise a TD algorithm which estimates ρ_d and uses it to find θ, which we call COP-TD(0, β) (Consistent Off-Policy TD).

Algorithm 1 COP-TD(0,β), Input: θ_0, $\widehat{\rho}_{d,0}$,

```
 1: Init: $F_0 = 0, \quad n_0^\beta = 1, \quad N(s) = 0$
 2: for $t = 1, 2, \ldots$ do
 3:     Observe $s_t, a_t, r_t, s_{t+1}$
 4:     Update normalization terms:
 5:     $N(s_t) = N(s_t) + 1, \quad \forall s \in \mathcal{S} : \hat{d}_\mu(s) = \frac{N(s)}{t}$
 6:     $n_t^\beta = \beta n_t^\beta + 1$
 7:     Update $\Gamma_t^n$'s weighted average:
 8:     $F_t = \rho_{t-1}(\beta F_{t-1} + e_{s_{t-1}})$
 9:     Update & project by $\rho_d$'s TD error:
10:     $\delta_t^d = \underbrace{\frac{F_t^\top \widehat{\rho}_{d,t}}{n_t^\beta}}_{\to \tilde{\Gamma}_t^\beta} - \widehat{\rho}_{d,t}(s_t)$
11:     $\widehat{\rho}_{d,t+1} = \Pi_{\Delta_{\hat{d}_\mu}} \left( \widehat{\rho}_{d,t} + \alpha_t^d \delta_t^d e_{s_t} \right)$
12:     Off-policy TD(0):
13:     $\delta_t = r_t + \theta_t^\top (\gamma\phi(s_{t+1}) - \phi(s_t))$
14:     $\theta_{t+1} = \theta_t + \alpha_t \widehat{\rho}_{d,t+1}(s_t) \rho_t \delta_t \phi(s_t)$
15: end for
```

Similarly to the Bellman operator for TD-learning, we define the underlying COP-operator Y and its β extension:

$$Yu = D_\mu^{-1} P_\pi^\top D_\mu u,$$
$$Y^\beta u = (1-\beta) D_\mu^{-1} P_\pi^\top (I - \beta P_\pi^\top)^{-1} D_\mu u.$$

The following Lemma may give some intuition on the convergence of the ρ_d estimation process:

Lemma 3. *Under the ergodicity assumption, denote the eigenvalues of P_π by $0 \le \cdots \le |\xi_2| < \xi_1 = 1$. Then Y^β is a $\max_{i \neq 1} \frac{(1-\beta)|\xi_i|}{|1-\beta\xi_i|} < 1$-contraction in the L_2-norm on the orthogonal subspace to ρ_d, and ρ_d is a fixed point of Y^β.*

The technical proof is given in the appendix.

Theorem 1. *If the step sizes satisfy $\sum_t \alpha_t = \sum_t \alpha_t^d = \infty, \sum_t (\alpha_t^2 + (\alpha_t^d)^2) < \infty, \frac{\alpha_t}{\alpha_t^d} \to 0, t\alpha_t^d \to 0$, and $\mathbb{E}\left[(\beta^n \Gamma_t^n)^2 | s_t \right] \le C$ for some constant C and every t and n, then after applying COP-TD(0, β), $\widehat{\rho}_{d,t}$ converges to ρ_d almost surely, and θ_t converges to the fixed point of $\Pi_\pi T_\pi V$.*

Notice that COP-TD(0, β) given in Alg. 1 is infeasible in problems with large state spaces since $\rho_d \in \mathbb{R}^{|\mathcal{S}|}$. Like TD($\lambda$), we can introduce linear function approximation: represent $\rho_d(s) \approx \theta_\rho^\top \phi_\rho(s)$ where θ_ρ is a weight vector and $\phi_\rho(s)$ is the off-policy feature vector and adjust the algorithm accordingly. For $\widehat{\rho_d}$ to still be contained in the set Δ_{d_μ}, we pose the requirement on the feature vectors: $\phi_\rho(s) \in \mathbb{R}_+^k$, and $\sum_s d_\mu(s) \theta_\rho^\top \phi_\rho(s) = 1$ $\big($noted as the simplex projection $\Pi_{\Delta_{\mathbb{E}_\mu[\phi_\rho(s)]}}\big)$. In practice, the latter requirement can be approximated: $\sum_s d_\mu(s) \theta_\rho^\top \phi_\rho(s) \approx \frac{1}{t} \theta_\rho^\top \sum_t \phi_\rho(s_t) = 1$ resulting in an extension of the previously applied d_μ estimation (step 5 in COP-TD(0, β)). We provide the full details in Algorithm 2, which also incorporates non-zero λ $\big($similarly to ETD(λ,β)$\big)$.

Algorithm 2 COP-TD(λ,β) with Function Approximation, Input: θ_0, $\theta_{\rho,0}$

```
 1: Init: $F_0 = \underline{0}, \quad n_0^\beta = 1, \quad N_\phi = \underline{0}, \quad e_0 = \underline{0}$
 2: for $t = 1, 2, \ldots$ do
 3:     Observe $s_t, a_t, r_t, s_{t+1}$
 4:     Update normalization terms:
 5:     $n_t^\beta = \beta n_t^\beta + 1, \quad N_\phi = N_\phi + \phi_\rho(s_t), \quad \hat{d}_{\phi_\rho} = \frac{N_\phi}{t}$
 6:     Update $\Gamma_t^n$'s weighted average:
 7:     $F_t = \rho_{t-1}(\beta F_{t-1} + \phi_\rho(s_{t-1}))$
 8:     Update & project by $\rho_d$'s TD error:
 9:     $\delta_t^d = \theta_{\rho,t-1}^\top \left( \frac{F_t}{n_t^\beta} - \phi_\rho(s_t) \right)$
10:     $\theta_{\rho,t+1} = \Pi_{\Delta_{\hat{d}_{\phi_\rho}}} \left( \theta_{\rho,t} + \alpha_t^d \delta_t^d \phi_\rho(s_t) \right)$
11:     Off-policy TD($\lambda$):
12:     $M_t = \lambda + (1-\lambda) \theta_{\rho,t+1}^\top \phi_\rho(s_t)$
13:     $e_t = \rho_t \left( \lambda\gamma e_t + M_t \phi(s_{t+1}) \right)$
14:     $\delta_t = r_t + \theta_t^\top (\gamma\phi(s_{t+1}) - \phi(s_t))$
15:     $\theta_{t+1} = \theta_t + \alpha_t \delta_t e_t$
16: end for
```

Theorem 2. *If the step sizes satisfy $\sum_t \alpha_t = \sum_t \alpha_t^d = \infty, \sum_t (\alpha_t^2 + (\alpha_t^d)^2) < \infty, \frac{\alpha_t}{\alpha_t^d} \to 0, t\alpha_t^d \to 0$, and $\mathbb{E}\left[(\beta^n \Gamma_t^n)^2 | s_t \right] \le C$ for some constant C and every t, n,* *then after applying COP-TD(0, β) with function approximation satisfying $\phi_\rho(s) \in \mathbb{R}^k_+$, $\widehat{\rho}_{d,t}$ converges to the fixed point of $\Pi_{\Delta_{\mathbb{E}_\mu[\phi_\rho]}}\Pi_{\phi_\rho} Y^\beta$ denoted by ρ_d^{COP} almost surely, and if θ_t converges it is to the fixed point of $\Pi_{d_\mu \circ \rho_d^{COP}} T_\pi V$, where $\circ$ is a coordinate-wise product of vectors.*

The proof is given in the appendix and also follows the ODE method. Notice that a theorem is only given for $\lambda = 0$, convergence results for general λ should follow the work by Yu (2015).

A possible criticism on COP-TD(0,β) is that it is not actually consistent, since in order to be consistent the original state space has to be small, in which case every off-policy algorithm is consistent as well. Still, the dependence on another set of features allows to trade-off accuracy with computational power in estimating ρ_d and subsequently V. Moreover, smart feature selection may further reduce this gap, and COP-TD(0, β) is still the first algorithm addressing this issue. We conclude with linking the error in ρ_d's estimate with the difference in the resulting θ, which suggests that a well estimated ρ_d results in consistency:

Corollary 1. *Let $0 < \epsilon < 1$. If $(1-\epsilon)\rho_d \le \rho_d^{COP} \le (1+\epsilon)\rho_d$, then the fixed point of COP-TD(0,β) with function approximation θ^{COP} satisfies the following, where $\|\cdot\|_\infty$ is the L_∞ induced norm:*

$$\|\theta^* - \theta^{COP}\|_\infty \le \epsilon \|A_\pi^{-1}\Phi^\top\|_\infty \left(R_{max} + (1+\gamma)\|\Phi\|_\infty \|\theta^{COP}\|_\infty\right),$$

where $A_\pi = \Phi^\top D_\pi (I - \gamma P_\pi)\Phi$, and θ^* sets the fixed point of the operator $\Pi_{d_\pi} T_\pi V$.

5.1. Relation to ETD(λ, β)

Recently, Sutton et al. (2015) had suggested an algorithm for off-policy evaluation called Emphatic TD. Their algorithm was later on extended by Hallak et al. (2015) and renamed ETD(λ, β), which was shown to perform extremely well empirically by White and White (2016). ETD(0, β) can be represented as:

$$\begin{aligned} F_t &= (1-\beta)\sum_{n=0}^{\infty} \beta^n \Gamma_t^n, \\ \theta_{t+1} &= \theta_t + \alpha_t F_t \rho_t \left(r_t + \theta_t^\top (\gamma\phi(s_{t+1}) - \phi(s_t))\right). \end{aligned} \tag{4}$$

As mentioned before, ETD(λ, β) converges to the fixed point of $\Pi_f T_\pi^\lambda$ (Yu, 2015), where $f = \mathbb{E}[F_t|s_t] = (I - \beta P_\pi)^{-1} d_\mu$. Error bounds can be achieved by showing that the operator $\Pi_f T_\pi^\lambda$ is a contraction under certain requirements on β and that the variance of F_t is directly related to β as well (Hallak et al., 2015) (and thus affects the convergence rate of the process).

When comparing ETD(λ,β)'s form to COP-TD(λ,β)'s, instead of spending memory and time resources on a state/feature-dependent F_t, ETD(λ,β) uses a one-variable approximation. The resulting F_t is in fact a one-step estimate of ρ_d, starting from $\widehat{\rho}_d(s) \equiv 1$ (see Equations 9, 4), up to a minor difference: $F_t^{\text{ETD}} = \beta F_t^{\text{COP-TD}} + 1$ (which following our logic adds bias to the estimate [2]).

Unlike ETD(λ, β), COP-TD(λ,β)'s effectiveness depends on the available resources. The number of features $\phi_\rho(s)$ can be adjusted accordingly to provide the most affordable approximation. The added cost is fine-tuning another step-size, though β's effect is less prominent.

6. The Logarithm Approach for Handling Long Products

We now present a heuristic algorithm which works similarly to COP-TD(λ, β). Before presenting the algorithm, we explain the motivation behind it.

6.1. Statistical Interpretation of TD(λ)

Konidaris et al. (2011) suggested a statistical interpretation of TD(λ). They show that under several assumptions the TD(λ) estimate $R_{s_t}^\lambda$ is the maximum likelihood estimator of $V(s_t)$ given $R_{s_t}^n$: (1) Each $R_{s_t}^n$ is an unbiased estimator of $V(s_t)$; (2) The random variables $R_{s_t}^n$ are independent and specifically uncorrelated; (3) The random variables $R_{s_t}^n$ are jointly normally distributed; and (4) The variance of each $R_{s_t}^n$ is proportional to λ^n.

Under Assumptions 1-3 the maximum likelihood estimator of $V(s)$ given its previous estimate can be represented as a linear convex combination of $R_{s_t}^n$ with weights:

$$w_n = \frac{\left[\text{Var}\left(R_{s_t}^{(n)}\right)\right]^{-1}}{\sum_{m=0}^{\infty}\left[\text{Var}\left(R_{s_t}^{(m)}\right)\right]^{-1}}.$$

Subsequently, in Konidaris et al. (2011) Assumption 4 was relaxed and instead a closed form approximation of the variance was proposed. In a follow-up paper by Thomas et al. (2015b), the second assumption was also removed and the weights were instead given as: $w_n = \frac{\mathbf{1}^\top cov(\mathbf{R}_{s_t})\mathbf{e}_n}{\mathbf{1}^\top cov(\mathbf{R}_{s_t})\mathbf{1}}$, where the covariance matrix can be estimated from the data, or otherwise learned through some parametric form.

While both the approximated variance and learned covariance matrix solutions improve performance on several benchmarks, the first uses a rather crude approximation, and the second solution is both state-dependent and based on noisy estimates of the covariance matrix. In addition, there aren't efficient on-line implementations since all past

[2] We have conducted several experiments with an altered ETD and indeed obtained better results compared with the original, these experiments are outside the scope of the paper.

weights should be recalculated to match a new sample. Still, the suggested statistical justification is a valuable tool in assessing the similar role of β in ETD(λ, β).

6.2. Variance Weighted Γ_t^n

As was shown by Konidaris et al. (2011), we can use state-dependent weights instead of β exponents to obtain better estimates. The second moments are given explicitly as follows[3]: $\mathbb{E}\left[(\Gamma_t^n)^2\,|s_t\right] = \frac{d_\mu^\top \tilde{P}^{n-1}\mathbf{e}_{s_t}}{d_\mu(s_t)}$, where $\left[\tilde{P}\right]_{s,s'} = \sum_{a\in\mathcal{A}} \frac{\pi^2(a|s)}{\mu(a|s)} P(s'|s,a)$.

These can be estimated for each state separately. Notice that the variances increase exponentially depending on the largest eigenvalue of $\tilde{P}$ (as Assumption 4 dictates), but this is merely an asymptotic behavior and may be relevant only when the weights are already negligible. Hence, implementing this solution on-line should not be a problem with the varying weights, as generally only the first few of these are non-zero. While this solution is impractical in problems with large state spaces parameterizing or approximating these variances (similarly to Thomas et al. (2015b)) could improve performance in specific applications.

6.3. Log-COP-TD(λ, β)

Assumption 3 in the previous section is that the sampled estimators $(R^{(n)}, \Gamma_t^n)$ are normally distributed. For on policy TD(λ), this assumption might seem not too harsh as the estimators $R^{(n)}$ represent growing **sums** of random variables. However, in our case the estimators Γ_t^n are growing **products** of random variables. To correct this issue we can define new estimators using a logarithm on each $\tilde{\Gamma}_t^n$:

$$\begin{aligned}\log\left[\rho_d(s_t)\right] &= \log\left[\mathbb{E}\left[\widehat{\rho_d}(s_{t-m})\prod_{k=t-m}^{t-1}\rho_k \;\middle|\; s_t\right]\right] \\ &\approx \log\left[\widehat{\rho_d}(s_{t-m})\right] + \sum_{k=t-m}^{t-1}\mathbb{E}\left[\log\left[\rho_k\right]|s_t\right].\end{aligned} \tag{5}$$

This approximation is crude – we could add terms reducing the error through Taylor expansion, but these would be complicated to deal with. Hence, we can relate to this method mainly as a well-motivated heuristic.

Notice that this formulation resembles the standard MDP formulation, only with the corresponding "reward" terms $\log[\rho_t]$ going backward instead of forward, and no discount factor. Unfortunately, without a discount factor we cannot expect the estimated value to converge, so we propose using an artificial one $\gamma_{\log}$. We can incorporate function approximation for this formulation as well. Unlike COP-TD(λ, β), we can choose the features and weights as we wish with no restriction, besides the linear constraint on the resulting ρ_d through the weight vector θ_ρ. This can be approximately enforced by normalizing θ_ρ using $\frac{X}{t} \doteq \frac{1}{t}\sum_t \exp(\theta_{\rho,t}^\top\phi(s_t))$ (which should equal 1 if we were exactly correct). We call the resulting algorithm Log-COP-TD(λ,β).

Algorithm 3 Log-COP-TD(λ,β) with Function Approximation, Input: θ_0,$\theta_{\rho,0}$

```
Init: $F_0 = 0, \quad n_0(\beta) = 1, \quad N(s) = 0$
for $t = 1, 2, ...$ do
    Observe $s_t, a_t, r_t, s_{t+1}$
    Update normalization terms:
    $n_t^\beta = \beta n_t^\beta + 1, \quad N_\phi = \gamma_{\log}(\beta N_\phi + \phi_\rho(s_t)), \quad X = X + \exp(\theta_{\rho,t}^\top\phi(s_t))$
    Update $\log(\Gamma_t^n)$'s weighted average:
    $F_t = \beta\gamma_{\log}F_{t-1} + n_t^\beta \log[\rho(s_{t-1})]$
    Update & project by $\log(\rho_d)$'s TD error:
    $\delta_t^d = \frac{F_t}{n_t^\beta} + \theta_{\rho,t}^\top\left(\frac{N_\phi}{n_t^\beta} - \phi_\rho(s_t)\right)$
    $\theta_{\rho,t+1} = \theta_{\rho,t} + \alpha_t^d\delta_t^d\phi_\rho(s_t)$
    Off-policy TD($\lambda$):
    $M_t = \lambda + (1-\lambda)\exp\left(\theta_{\rho,t+1}^\top\phi_\rho(s_t)\right)/(X/t)$
    $e_t = \rho_t\left(\lambda\gamma e_t + M_t\phi(s_{t+1})\right)$
    $\delta_t = r_t + \theta_t^\top(\gamma\phi(s_{t+1}) - \phi(s_t))$
    $\theta_{t+1} = \theta_t + \alpha_t\delta_t e_t$
end for
```

6.4. Using the Original Features

An interesting phenomenon occurs when the behavior and target policies employ a feature based Boltzmann distribution for choosing the actions: $\mu(a|s) = \exp\left(\theta_{a,\mu}^\top\phi(s)\right)$, and $\pi(a|s) = \exp\left(\theta_{a,\pi}^\top\phi(s)\right)$, where a constant feature is added to remove the (possibly different) normalizing constant. Thus, $\log(\rho_t) = (\theta_{a,\pi} - \theta_{a,\mu})^\top\phi(s_t)$, and Log-COP-TD($\lambda$,$\beta$) obtains a parametric form that depends on the original features instead of a different set.

6.5. Approximation Hardness

As we propose to use linear function approximation for $\rho_d(s)$ and $\log\left(\rho_d(s)\right)$ one cannot help but wonder how hard it is to approximate these quantities, especially compared to the value function. The comparison between $V(s)$ and $\rho_d(s)$ is problematic for several reasons:

1. The ultimate goal is estimating $V^\pi(s)$, approximation errors in $\rho_d(s)$ are second order terms.

2. The value function $V^\pi(s)$ depends on the policy-

[3] The covariances can be expressed analytically as well, for clarity we drop this immediate result.

induced reward function and transition probability matrix, while $\rho_d(s)$ depends on the stationary distributions induced by both policies. Since each depends on at least one distinct factor - we can expect different setups to result in varied approximation hardness. For example, if the reward function has a poor approximation then so will $V^\pi(s)$, while extremely different behavior and target policies can cause $\rho_d(s)$ to behave erratically.

3. Subsequently, the choice of features for approximating $V^\pi(s)$ and $\rho_d(s)$ can differ significantly depending on the problem at hand.

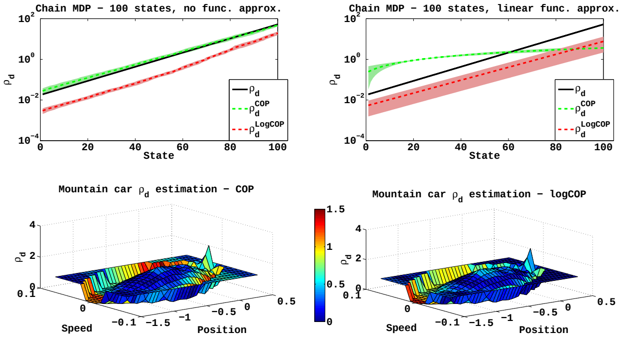
If we would still like to compare $V^\pi(s)$ and $\rho_d(s)$, we could think of extreme examples:

- When $\pi = \mu$, $\rho_d(s) \equiv 1$, when $R(s) \equiv 0$ then $V^\pi(s) \equiv 0$.
- In the chain MDP example in Section 4 we saw that $\rho_d(s)$ is an exponential function of the location in the chain. Setting reward in one end to 1 will result in an exponential form for $V^\pi(s)$ as well. Subsequently, in the chain MDP example approximating $\log(\rho_d(s))$ is easier than $\rho_d(s)$ as we obtain a linear function of the position; This is not the general case.

7. Experiments

We have performed 3 types of experiments. Our first batch of experiments (Figure 1) demonstrates the accuracy of predicting ρ_d by both COP-TD(λ, β) and Log-COP-TD(λ, β). We show two types of setups in which visualization of ρ_d is relatively clear - the chain MDP example mentioned in Section 4 and the mountain car domain (Sutton and Barto, 1998) in which the state is determined by only two continuous variables - the car's position and speed. The parameters λ and β exhibited low sensitivity in these tasks so they were simply set to 0, we show the estimated ρ_d after 10^6 iterations. For the chain MDP (top two plots, notice the logarithmic scale) we first approximate ρ_d without any function approximation (top-left) and we can see COP-TD manages to converge to the correct value while Log-COP-TD is much less exact. When we use linear feature space (constant parameter and position) Log-COP-TD captures the true behavior of ρ_d much better as expected. The two lower plots show the error (in color) in ρ_d estimated for the mountain car with a pure exploration behavior policy vs. a target policy oriented at moving right. The z-axis is the same for both plots and it describes a much more accurate estimate of ρ_d obtained through simulations. The features used were local state aggregation. We can see that both algorithms succeed similarly on the position-speed pairs which are sampled often due to the behavior policy and the mountain. When looking at more rarely observed states, the estimate becomes worse for both algorithms, though Log-COP-TD seems to be better performing on the spike at position > 0.

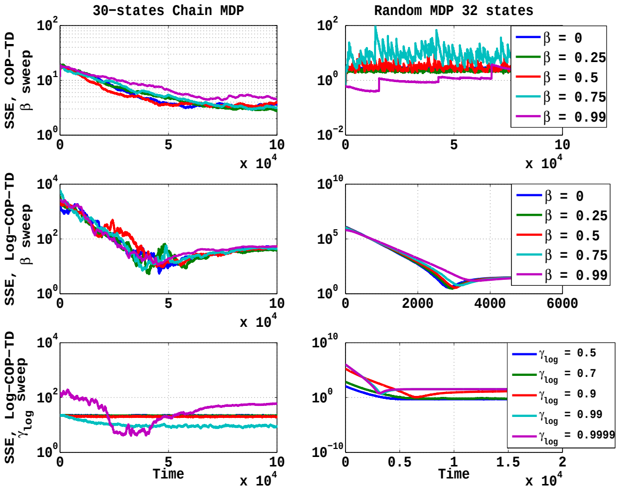
Figure 1. Estimation quality of COP-TD and Log-COP-TD in the chain MDP (top) and mountain car (bottom) problems. The chain MDP plots differ by the function approximation and the shading reflects one standard deviation over 10 trajectories. The mountain car plots compare COP-TD with Log-COP-TD where the z-axis is the same (true ρ_d) with the colors specifying the error.



Next we test the sensitivity of COP-TD(λ, β) and Log-COP-TD(λ,β) to the parameters β and $\gamma_{\log}$ (Figure 2) on two distinct toy examples - the chain MDP introduced before but with only 30 states with the position-linear features, and a random MDP with 32 states, 2 actions and a 5-bit binary feature vector along with a free parameter (this compact representation was suggested by White and White (2016) to approximate real world problems). The policies on the chain MDP were taken as described before, and on the random MDP a state independent 0.75/0.25 probability to choose an action by the behavior/target policy. As we can see, larger values of β cause noisier estimations in the random MDP for COP-TD(λ, β), but has little effect in other venues. As for $\gamma_{\log}$ - we can see that if it is too large or too small the error behaves sub-optimally, as expected for the crude approximation of Equation 5. In conclusion, unlike ETD(λ, β), Log/COP-TD(λ, β) are much less effected by β, though $\gamma_{\log}$ should be tuned to improve results.

Our final experiment (Figure 3) compares our algorithms to ETD(λ, β) and GTD(λ, β) over 4 setups: chain MDP with 100 states with right half rewards 1 with linear features, a 2 action random MDP with 256 states and binary features, acrobot (3 actions) and cart-pole balancing (21 actions) (Sutton and Barto, 1998) with reset at success and state aggregation to 100 states. In all problems we used the same features for ρ_d and $V^\pi(s)$ estimation, $\gamma = 0.99$, constant step size 0.05 for the TD process and results were averaged over 10 trajectories, other parameters (λ, β, other step sizes, $\gamma_{\log}$) were swiped over to find the best ones. To

Figure 2. The effect of $\beta, \gamma_{\log}$ on COP-TD(λ,β) and Log-COP-TD(λ,β), the y-axis is ρ_d's estimation sum of squared errors (SSE) over all states.



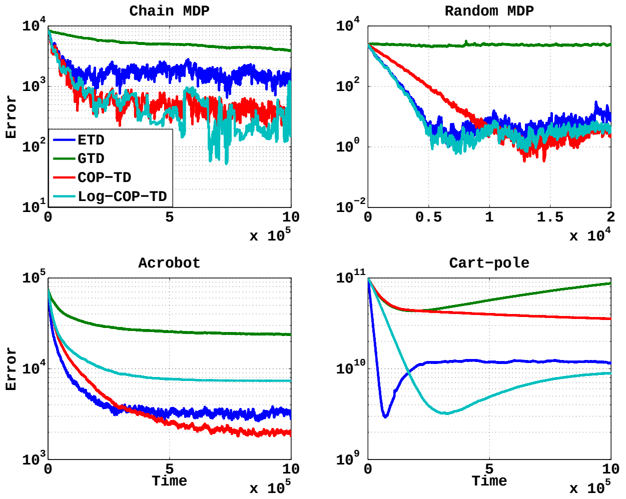
reduce figure clutter we have not included standard deviations though the noisy averages still reflect the variance in the process. Our method of comparison on the first 2 setups estimates the value function using the suggested algorithm, and finds the d_π weighted average of the error between V and the on-policy fixed point $\Pi_\pi T V_\pi$:

$$\|\hat{V} - \Pi_\pi T V_\pi\|_{d_\pi}^2 = \sum_s d_\pi(s) \left[(\theta^* - \hat{\theta})^\top \phi(s) \right]^2 ,$$

where θ^* is the optimal θ obtained by on-policy TD using the target policy. On the latter continuous state problems we applied on-line TD on a different trajectory following the target policy, used the resulting θ value as ground truth and taken the sum of squared errors with respect to it. The behavior and target policies for the chain MDP and random MDP are as specified before. For the acrobot problem the behavior policy is uniform over the 3 actions and the target policy chooses between these with probabilities $(\frac{1}{6}, \frac{1}{3}, \frac{1}{2})$. For the cart-pole the action space is divided to 21 actions from -1 to 1 equally, the behavior policy chooses among these uniformly while the target policy is 1.5 times more prone to choosing a positive action than a negative one.

The experiments show that COP-TD(λ, β) and Log-COP-TD(λ, β) have comparable performance to ETD(λ, β) where at least one is better in every setup. The advantage in the new algorithms is especially seen in the chain MDP corresponding to a large discrepancy between the stationary distribution of the behavior and target policy. GTD(λ) is consistently worse on the tested setups, this might be due to the large difference between the chosen behavior and target policies which affects GTD(λ) the most.

Figure 3. Error over time of several on-line off-policy algorithms.



8. Conclusion

Research on off-policy evaluation has flourished in the last decade. While a plethora of algorithms were suggested so far, ETD(λ, β) by Hallak et al. (2015) has perhaps the simplest formulation and theoretical properties. Unfortunately, ETD(λ, β) does not converge to the same point achieved by on-line TD when linear function approximation is applied.

We address this issue with COP-TD(λ,β) and proved it can achieve consistency when used with a correct set of features, or at least allow trading-off some of the bias by adding or removing features. Despite requiring a new set of features and calibrating an additional update function, COP-TD(λ,β)'s performance does not depend as much on β as ETD(λ,β), and shows promising empirical results.

We offer a connection to the statistical interpretation of TD(λ) that motivates our entire formulation. This interpretation leads to two additional approaches: (a) weight the Γ_t^n using estimated variances instead of β exponents and (b) approximating $\log[\rho_d]$ instead of ρ_d; both approaches deserve consideration when facing a real application.

9. Acknowledgments

This Research was supported in part by the Israel Science Foundation (grant No. 920/12) and by the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013)/ ERC Grant Agreement n.306638.

References

Hasan AA Al-Rawi, Ming Ann Ng, and Kok-Lim Alvin Yau. Application of reinforcement learning to routing in distributed wireless networks: a review. *Artificial Intelligence Review*, 43(3):381–416, 2015.

Enda Barrett, Enda Howley, and Jim Duggan. Applying reinforcement learning towards automating resource allocation and application scalability in the cloud. *Concurrency and Computation: Practice and Experience*, 25 (12):1656–1674, 2013.

D. Bertsekas. *Dynamic Programming and Optimal Control, Vol II*. Athena Scientific, 4th edition, 2012.

D. Bertsekas and J. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

D. Bertsekas and H. Yu. Projected equation methods for approximate solution of large linear systems. *Journal of Computational and Applied Mathematics*, 227(1):27–50, 2009.

Shalabh Bhatnagar, Vivek S Borkar, and LA Prashanth. Adaptive feature pursuit: Online adaptation of features in reinforcement learning. *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*, pages 517–534, 2012.

Shalabh Bhatnagar, Vivek S Borkar, and KJ Prabuchandran. Feature search in the grassmanian in online reinforcement learning. *IEEE Journal of Selected Topics in Signal Processing*, 7(5):746–758, 2013.

Wendelin Böhmer, Steffen Grünewälder, Yun Shen, Marek Musial, and Klaus Obermayer. Construction of approximation spaces for reinforcement learning. *Journal of Machine Learning Research*, 14(1):2067–2118, 2013.

Justin A Boyan. Least-squares temporal difference learning. In *ICML*, pages 49–56, 1999.

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

Yunmei Chen and Xiaojing Ye. Projection onto a simplex. *arXiv preprint arXiv:1101.6081*, 2011.

Christoph Dann, Gerhard Neumann, and Jan Peters. Policy evaluation with temporal differences: a survey and comparison. *Journal of Machine Learning Research*, 15(1): 809–883, 2014.

Dotan Di Castro and Shie Mannor. Adaptive bases for reinforcement learning. *Machine Learning and Knowledge Discovery in Databases*, pages 312–327, 2010.

Amir M Farahmand, Mohammad Ghavamzadeh, Shie Mannor, and Csaba Szepesvári. Regularized policy iteration. In *Advances in Neural Information Processing Systems*, pages 441–448, 2009.

Clement Gehring, Yangchen Pan, and Martha White. Incremental truncated lstd. *arXiv preprint arXiv:1511.08495*, 2015.

Matthieu Geist and Bruno Scherrer. l1-penalized projected bellman residual. In *European Workshop on Reinforcement Learning*, pages 89–101. Springer, 2011.

Matthieu Geist and Bruno Scherrer. Off-policy learning with eligibility traces: A survey. *The Journal of Machine Learning Research*, 15(1):289–333, 2014.

Matthieu Geist, Bruno Scherrer, Alessandro Lazaric, and Mohammad Ghavamzadeh. A dantzig selector approach to temporal difference learning. *arXiv preprint arXiv:1206.6480*, 2012.

Mohammad Ghavamzadeh, Alessandro Lazaric, Odalric Maillard, and Rémi Munos. Lstd with random projections. In *Advances in Neural Information Processing Systems*, pages 721–729, 2010.

Sertan Girgin and Philippe Preux. Basis expansion in natural actor critic methods. In *European Workshop on Reinforcement Learning*, pages 110–123. Springer, 2008.

Arash Givchi and Maziar Palhang. Off-policy temporal difference learning with distribution adaptation in fast mixing chains. *Soft Computing*, pages 1–14, 2017.

Hirotaka Hachiya and Masashi Sugiyama. Feature selection for reinforcement learning: Evaluating implicit state-reward dependency via conditional mutual information. *Machine Learning and Knowledge Discovery in Databases*, pages 474–489, 2010.

Hirotaka Hachiya, Masashi Sugiyama, and Naonori Ueda. Importance-weighted least-squares probabilistic classifier for covariate shift adaptation with application to human activity recognition. *Neurocomputing*, 80:93–101, 2012.

Assaf Hallak, Aviv Tamar, Remi Munos, and Shie Mannor. Generalized emphatic temporal difference learning: Bias-variance analysis. *arXiv preprint arXiv:1509.05172*, 2015.

Irit Hochberg, Guy Feraru, Mark Kozdoba, Shie Mannor, Moshe Tennenholtz, and Elad Yom-Tov. Encouraging physical activity in patients with diabetes through automatic personalized feedback via reinforcement learning improves glycemic control. *Diabetes care*, 39(4):e59–e60, 2016.

Matthew W Hoffman, Alessandro Lazaric, Mohammad Ghavamzadeh, and Rémi Munos. Regularized least squares temporal difference learning with nested l2 and l1 penalization. In *European Workshop on Reinforcement Learning*, pages 102–114. Springer, 2011.

Jeff Johns and Sridhar Mahadevan. Constructing basis functions from directed graphs for value function approximation. In *Proceedings of the 24th international conference on Machine learning*, pages 385–392. ACM, 2007.

Jeffrey Johns, Christopher Painter-Wakefield, and Ronald Parr. Linear complementarity for regularized policy evaluation and improvement. In *Advances in neural information processing systems*, pages 1009–1017, 2010.

Takafumi Kanamori, Shohei Hido, and Masashi Sugiyama. A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, 10(Jul): 1391–1445, 2009.

Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, page 0278364913495721, 2013.

J Zico Kolter. The fixed points of off-policy TD. In *NIPS*, 2011.

J Zico Kolter and Andrew Y Ng. Regularization and feature selection in least-squares temporal difference learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 521–528. ACM, 2009.

George Konidaris, Scott Niekum, and Philip S Thomas. Td-gamma: Re-evaluating complex backups in temporal difference learning. In *Advances in Neural Information Processing Systems 24*, pages 2402–2410. Curran Associates, Inc., 2011. URL http://papers.nips.cc/paper/4472-td_gamma-re-evaluating-complex-backups-in-temporal-difference-learning.pdf.

Mark Kroon and Shimon Whiteson. Automatic feature selection for model-based reinforcement learning in factored mdps. In *Machine Learning and Applications, 2009. ICMLA'09. International Conference on*, pages 324–330. IEEE, 2009.

Harold Kushner and G George Yin. *Stochastic approximation and recursive algorithms and applications*, volume 35. Springer Science & Business Media, 2003.

Bo Liu, Sridhar Mahadevan, and Ji Liu. Regularized off-policy td-learning. In *Advances in Neural Information Processing Systems*, pages 836–844, 2012.

De-Rong Liu, Hong-Liang Li, and Ding Wang. Feature selection and feature learning for high-dimensional batch reinforcement learning: a survey. *International Journal of Automation and Computing*, 12(3):229–242, 2015.

Manuel Loth, Manuel Davy, and Philippe Preux. Sparse temporal difference learning using lasso. In *Approximate Dynamic Programming and Reinforcement Learning, 2007. ADPRL 2007. IEEE International Symposium on*, pages 352–359. IEEE, 2007.

Sridhar Mahadevan. Samuel meets amarel: Automating value function approximation using global state space analysis. In *AAAI*, volume 5, pages 1000–1005, 2005.

Sridhar Mahadevan and Bo Liu. Sparse q-learning with mirror descent. *arXiv preprint arXiv:1210.4893*, 2012.

Sridhar Mahadevan and Mauro Maggioni. Proto-value functions: A laplacian framework for learning representation and control in markov decision processes. *Journal of Machine Learning Research*, 8(Oct):2169–2231, 2007.

Sridhar Mahadevan et al. Learning representation and control in markov decision processes: New frontiers. *Foundations and Trends® in Machine Learning*, 1(4):403–565, 2009.

A Rupam Mahmood and Richard S Sutton. Off-policy learning based on weighted importance sampling with linear computational complexity. In *Conference on Uncertainty in Artificial Intelligence*, 2015.

A Rupam Mahmood, Hado P van Hasselt, and Richard S Sutton. Weighted importance sampling for off-policy learning with linear function approximation. In *Advances in Neural Information Processing Systems*, pages 3014–3022, 2014.

Ishai Menache, Shie Mannor, and Nahum Shimkin. Basis function adaptation in temporal difference reinforcement learning. *Annals of Operations Research*, 134(1):215–238, 2005.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Christopher Painter-Wakefield and Ronald Parr. Greedy algorithms for sparse reinforcement learning. *arXiv preprint arXiv:1206.6485*, 2012.

Ronald Parr, Christopher Painter-Wakefield, Lihong Li, and Michael Littman. Analyzing feature generation for value-function approximation. In *Proceedings of* *the 24th international conference on Machine learning*, pages 737–744. ACM, 2007.

Ronald Parr, Lihong Li, Gavin Taylor, Christopher Painter-Wakefield, and Michael L Littman. An analysis of linear models, linear value-function approximation, and feature selection for reinforcement learning. In *Proceedings of the 25th international conference on Machine learning*, pages 752–759. ACM, 2008.

Marek Petrik. An analysis of laplacian methods for value function approximation in mdps. In *IJCAI*, pages 2574–2579, 2007.

Marek Petrik, Gavin Taylor, Ron Parr, and Shlomo Zilberstein. Feature selection using regularization in approximate linear programs for markov decision processes. *arXiv preprint arXiv:1005.1860*, 2010.

Doina Precup, Richard S Sutton, and Sanjoy Dasgupta. Off-policy temporal-difference learning with function approximation. In *ICML*, 2001.

Zhiwei Qin, Weichang Li, and Firdaus Janoos. Sparse reinforcement learning via convex optimization. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 424–432, 2014.

Zeev Schuss and Vivek S Borkar. Stochastic approximation: A dynamical systems viewpoint, 2009.

William D Smart. Explicit manifold representations for value-function approximation in reinforcement learning. In *ISAIM*, 2004.

Yi Sun, Mark Ring, Jürgen Schmidhuber, and Faustino J Gomez. Incremental basis construction from temporal difference error. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 481–488, 2011.

R. S. Sutton and A. Barto. *Reinforcement learning: An introduction*. Cambridge Univ Press, 1998.

R. S. Sutton, A. R. Mahmood, and M White. An emphatic approach to the problem of off-policy temporal-difference learning. *arXiv:1503.04269*, 2015.

Rich Sutton, Ashique R Mahmood, Doina Precup, and Hado V Hasselt. A new q (lambda) with interim forward view and monte carlo equivalence. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 568–576, 2014.

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

Richard S Sutton, Hamid R Maei, and Csaba Szepesvári. A convergent o(n) temporal-difference algorithm for off-policy learning with linear function approximation. In *Advances in neural information processing systems*, pages 1609–1616, 2009a.

Richard S Sutton, Hamid Reza Maei, Doina Precup, Shalabh Bhatnagar, David Silver, Csaba Szepesvári, and Eric Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 993–1000. ACM, 2009b.

Georgios Theocharous and Assaf Hallak. Lifetime value marketing using reinforcement learning. *RLDM 2013*, page 19, 2013.

Georgios Theocharous, Philip S Thomas, and Mohammad Ghavamzadeh. Personalized ad recommendation systems for life-time value optimization with guarantees. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI-15)*, 2015.

Philip Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High confidence policy improvement. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 2380–2388, 2015a.

Philip S Thomas, Scott Niekum, Georgios Theocharous, and George Konidaris. Policy evaluation using the omega-return. In *Advances in Neural Information Processing Systems 28*, pages 334–342. Curran Associates, Inc., 2015b. URL http://papers.nips.cc/paper/5807-policy-evaluation-using-the-return.pdf.

John N Tsitsiklis and Benjamin Van Roy. An analysis of temporal-difference learning with function approximation. *Automatic Control, IEEE Transactions on*, 42(5): 674–690, 1997.

Hado van Hasselt, A Rupam Mahmood, and Richard S Sutton. Off-policy td (λ) with a true online equivalence. In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence, Quebec City, Canada*, 2014.

Jian Wang, Zhenhua Huang, and Xin Xu. A novel approach for constructing basis functions in approximate dynamic programming for feedback control. In *Adaptive Dynamic Programming And Reinforcement Learning (ADPRL), 2013 IEEE Symposium on*, pages 47–51. IEEE, 2013.

Adam White and Martha White. Investigating practical, linear temporal difference learning. *arXiv preprint arXiv:1602.08771*, 2016.

Dean S Wookey and George D Konidaris. Regularized feature selection in reinforcement learning. *Machine Learning*, 100(2-3):655–676, 2015.

Dean Stephen Wookey. *Representation discovery using a fixed basis in reinforcement learning*. PhD thesis, University of the Witwatersrand South Africa, 2016.

H. Yu. On convergence of emphatic temporal-difference learning. In *COLT*, 2015.

Huizhen Yu. Convergence of least squares temporal difference methods under general conditions. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 1207–1214, 2010.

Tom Zahavy, Nir Ben-Zrihem, and Shie Mannor. Graying the black box: Understanding dqns. *arXiv preprint arXiv:1602.02658*, 2016.