Emphatic Temporal-Difference Learning

A. Rupam Mahmood **Huizhen Yu** **Martha White** **Richard S. Sutton**
Reinforcement Learning and Artificial Intelligence Laboratory
Department of Computing Science, University of Alberta, Edmonton, AB T6G 2E8 Canada

Abstract

Emphatic algorithms are temporal-difference learning algorithms that change their effective state distribution by selectively emphasizing and de-emphasizing their updates on different time steps. Recent works by Sutton, Mahmood and White (2015), and Yu (2015) show that by varying the emphasis in a particular way, these algorithms become stable and convergent under off-policy training with linear function approximation. This paper serves as a unified summary of the available results from both works. In addition, we demonstrate the empirical benefits from the flexibility of emphatic algorithms, including state-dependent discounting, state-dependent bootstrapping, and the user-specified allocation of function approximation resources.

Keywords: temporal-difference learning, function approximation, off-policy learning, stability, convergence

1. Introduction

A fundamental problem in reinforcement learning involves learning a sequence of long-term predictions in a dynamical system. This problem is often formulated as learning approximations to value functions of Markov decision processes (Bertsekas & Tsitsiklis 1996, Sutton & Barto 1998). Temporal-difference learning algorithms, such as TD(λ) (Sutton 1988), GQ(λ) (Maei & Sutton 2010), and LSTD(λ) (Boyan 1999, Bradtke & Barto 1996), provide effective solutions to this problem. These algorithms stand out particularly because of their ability to learn efficiently on a moment-by-moment basis using memory and computational complexity that is constant in time. These methods are also distinguished due to their ability to learn from other predictions, a technique known as *bootstrapping*, which often provides fast and more accurate answers (Sutton 1988).

TD algorithms conventionally make updates at every state visited, implicitly giving higher importance, in terms of function-approximation resources, to states that are visited more frequently. As the value cannot be estimated accurately under function approximation, valuing some states more means valuing others less. We may, however, be interested in valuing some states more than others based on criteria other than visitation frequency. Conventional TD updates do not provide that flexibility and cannot be naively modified. For example, in the case of off-policy TD updates, updating according to one policy while learning about another can cause divergence (Baird 1995).

In this paper, we discuss emphatic TD(λ) (Sutton et al. 2015), a principled solution for the problem of selective updating, where convergence is ensured under an arbitrary interest in visited states as well as off-policy training. The idea is to emphasize and de-emphasize state updates with user-specific interest in conjunction with how much other states bootstrap from that state. We first describe this idea in a simpler case: linear function approximation with full bootstrapping (i.e., $\lambda = 0$). We then derive the full algorithm for the more general off-policy learning setting with arbitrary bootstrapping. Finally, after briefly summarizing the available results on the stability and convergence of the new algorithm, we discuss the use and the potential advantages of this algorithm using an illustrative experiment.

2. The problem of selective updates

Let us start with the problem of selective updating in the simplest function approximation case: linear TD(λ) with $\lambda = 0$. Consider a Markov decision process (MDP) with a finite set $\mathcal{S}$ of N states and a finite set $\mathcal{A}$ of actions, for the discounted total reward criterion with discount rate $\gamma \in [0, 1)$. In this setting, an agent interacts with the environment by taking an action $A_t \in \mathcal{A}$ at state $S_t \in \mathcal{S}$ according to a policy $\pi : \mathcal{A} \times \mathcal{S} \to [0, 1]$ where $\pi(a|s) \doteq \mathbb{P}\{A_t = a|S_t = s\}$[1], transitions to state $S_{t+1} \in \mathcal{S}$, and receives reward $R_{t+1} \in \mathbb{R}$ in a sequence of time steps $t \geq 0$. Let $\mathbf{P}_\pi \in \mathbb{R}^{N\times N}$ denote the state transition probability matrix and $\mathbf{r}_\pi \in \mathbb{R}^N$ the expected immediate rewards from each state under π. The value of a state is then defined as:

$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t|S_t = s]\,, \tag{1}$$

where $\mathbb{E}_\pi[\cdot]$ denotes an expectation conditional on all actions being selected according to π, and G_t, the *return* at time t, is a random variable of the future outcome:

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots\,. \tag{2}$$

We approximate the value of a state as a linear function of its features: $\boldsymbol{\theta}^\top \boldsymbol{\phi}(s) \approx v_\pi(s)$, where $\boldsymbol{\phi}(s) \in \mathbb{R}^n$ is the feature vector corresponding to state s. Conventional linear TD(0) learns the value function v_π by generating a sequence of parameter vectors $\boldsymbol{\theta}_t \in \mathbb{R}^n$:

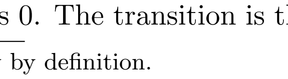
$$\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha\left(R_{t+1} + \gamma\boldsymbol{\theta}_t^\top\boldsymbol{\phi}(S_{t+1}) - \boldsymbol{\theta}_t^\top\boldsymbol{\phi}(S_t)\right)\boldsymbol{\phi}(S_t), \tag{3}$$

where $\alpha > 0$ is a step-size parameter.

Additionally, we may have a relative interest in each state, denoted by a nonnegative *interest function* $i : \mathcal{S} \to [0, \infty)$. For example, in episodic problems we often care primarily about the value of the first state, or of earlier states generally (Thomas 2014). A straightforward way to incorporate the relative interests into TD(0) would be to use $i(S_t)$ as a factor to the update on each state S_t:

$$\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha\left(R_{t+1} + \gamma\boldsymbol{\theta}_t^\top\boldsymbol{\phi}(S_{t+1}) - \boldsymbol{\theta}_t^\top\boldsymbol{\phi}(S_t)\right)i(S_t)\boldsymbol{\phi}(S_t). \tag{4}$$

In order to illustrate the problem of this approach, suppose there is a Markov chain consisting of two non-terminal and a terminal state with features $\boldsymbol{\phi}(1) = 1$ and $\boldsymbol{\phi}(2) = 2$ and interests $i(1) = 1$ and $i(2) = 0$ (cf. Tsitsiklis & Van Roy 1996):



Then the estimated values are θ and 2θ for a scalar parameter $\theta \in \mathbb{R}$. Suppose that θ is 10, the reward on the first transition is 0. The transition is then from a state valued at 10 to a

1. The notation $\doteq$ indicates an equality by definition.

state valued at 20. If $\gamma = 1$ and α is 0.1, then θ will be increased to 11. But then the next time the transition occurs there will be an even bigger increase in value, from 11 to 22, and a bigger increase in θ, to 12.1. If this transition is experienced repeatedly on its own, then the system is unstable and the parameter increases without bound—it diverges.

This problem arises due to both bootstrapping and the use of function approximation, which entails shared resources among the states. If a tabular representation was used instead, the value of each state would be stored independently and divergence would not occur. Likewise, if the value estimate of the first state was updated without bootstrapping from that of the second state, such divergence could again be avoided.

Emphatic TD(0) (Sutton et al. 2015) remedies this problem of TD(0) by emphasizing the update of a state, depending on how much a state is bootstrapped in conjunction with the relative interest in that state. Although $\lambda = 0$ gives full bootstrapping, the amount of bootstrapping is still modulated by γ. For example, if $\gamma = 0$, then no bootstrapping occurs even with $\lambda = 0$. The amount of emphasis to the update of a state at time t is:

$$F_t \doteq i(S_t) + \gamma i(S_{t-1}) + \gamma^2 i(S_{t-2}) + \cdots + \gamma^t i(S_0) = i(S_t) + \gamma F_{t-1}. \tag{5}$$

The following update defines emphatic TD(0):

$$\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha \left(R_{t+1} + \gamma \boldsymbol{\theta}_t^\top \boldsymbol{\phi}(S_{t+1}) - \boldsymbol{\theta}_t^\top \boldsymbol{\phi}(S_t) \right) F_t \boldsymbol{\phi}(S_t).$$

According to this algorithm, the value estimate of a state is updated if the user is interested in that state or it is reachable from another state in which the user is interested. Going back to the above two-state example, the second state value is now also updated despite having a user-specified interest of 0. In fact, F_t is equal for both states, and updating is exactly equivalent to on-policy sampling; hence, divergence does not occur. For other choices of relative interest and discount rate, the effective state distribution can be different than on-policy sampling, but the algorithm still converges as we show later.

3. ETD(λ): The off-policy emphatic TD(λ)

In this section, we develop the emphatic TD algorithm, which we call *ETD(λ),* in the generic setting of off-policy training with state-dependent discounting and bootstrapping.

Let $\gamma : \mathcal{S} \to [0, 1]$ be the state-dependent degree of discounting; equivalently, $1 - \gamma(s)$ is the probability of terminating upon arrival in state s. Let $\lambda : \mathcal{S} \to [0, 1]$ denote a state-dependent degree of bootstrapping; in particular, $1 - \lambda(s)$ determines the degree of bootstrapping upon arriving in state s. As notational shorthand, we use $\gamma_t \doteq \gamma(S_t)$, $\lambda_t \doteq \lambda(S_t)$, and $\boldsymbol{\phi}_t \doteq \boldsymbol{\phi}(S_t)$. For TD learning, we define a general notion of bootstrapped return, the λ*-return*, with state-dependent bootstrapping and discounting, by

$$G_t^\lambda \doteq R_{t+1} + \gamma_{t+1} \left((1 - \lambda_{t+1}) \boldsymbol{\theta}_t^\top \boldsymbol{\phi}_{t+1} + \lambda_{t+1} G_{t+1}^\lambda \right).$$

This return can be directly used to estimate v_π on-policy as long as the agent follows π. However, in off-policy learning, experience is generated by following a different policy $\mu : \mathcal{A} \times \mathcal{S} \to [0, 1]$, often called the *behavior* policy. To obtain an unbiased estimate of the return under π, the experience generated under μ has to be reweighted by importance sampling ratios: $\rho_t \doteq \frac{\pi(A_t|S_t)}{\mu(A_t|S_t)}$, assuming $\mu(a|s) > 0$ for every state and action for which $\pi(a|s) > 0$. The importance-sampled λ-return for off-policy learning is thus defined as follows (Maei 2011, van Hasselt et al. 2014):

$$G_t^{\lambda\rho} \doteq \rho_t \left(R_{t+1} + \gamma_{t+1} \left((1 - \lambda_{t+1}\rho_{t+1}) \boldsymbol{\theta}_t^\top \boldsymbol{\phi}_{t+1} + \lambda_{t+1} G_{t+1}^{\lambda\rho} \right) \right).$$

The forward-view update of the conventional off-policy TD(λ) can be written as:

$$\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha \left(G_t^{\lambda\rho} - \rho_t \boldsymbol{\phi}_t^\top \boldsymbol{\theta}_t \right) \boldsymbol{\phi}_t. \tag{6}$$

The backward-view update with an offline equivalence (cf. van Seijen & Sutton 2014) with the above forward view can be written as:

$$\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha \left(R_{t+1} + \gamma_{t+1} \boldsymbol{\theta}_t^\top \boldsymbol{\phi}_{t+1} - \boldsymbol{\theta}_t^\top \boldsymbol{\phi}_t \right) \mathbf{e}_t \tag{7}$$

$$\mathbf{e}_t \doteq \rho_t \left(\gamma_t \lambda_t \mathbf{e}_{t-1} + \boldsymbol{\phi}_t \right), \quad \text{with } \mathbf{e}_{-1} \doteq \mathbf{0}, \tag{8}$$

where $\mathbf{e}_t \in \mathbb{R}^n$ is the eligibility-trace vector at time t. This algorithm makes an update to each state visited under μ and does not allow user-specified relative interests to different states. Convergence is also not guaranteed in general for this update rule.

By contrast, instead of (6), we define the forward view of ETD(λ) to be:

$$\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha \left(G_t^{\lambda\rho} - \rho_t \boldsymbol{\phi}_t^\top \boldsymbol{\theta}_t \right) M_t \boldsymbol{\phi}_t. \tag{9}$$

Here $M_t \in \mathbb{R}$ denotes the emphasis given to update at time t, and it is derived based on the following reasoning, similar to the derivation of F_t for emphatic TD(0).

The emphasis to the update at state S_t is first and foremost, due to $i(S_t)$, the inherent interest of the user to that state. A portion of the emphasis is also due to the amount of bootstrapping the preceding state S_{t-1} does from S_t, determined by $\gamma_t(1-\lambda_t)\rho_{t-1}$: the probability of not terminating at S_t times the probability of bootstrapping at S_t times the degree by which the preceding transition is followed under the target policy. Finally, M_t also depends on M_{t-1}, the emphasis of the preceding state itself. The emphasis for state S_t similarly depends on all the preceding states that bootstrap from this state to some extent. Thus the total emphasis can be written as:

$$M_t \doteq i(S_t) + \sum_{k=0}^{t-1} M_k \rho_k \left(\prod_{i=k+1}^{t-1} \gamma_i \lambda_i \rho_i \right) \gamma_t (1 - \lambda_t) = \lambda_t i(S_t) + (1 - \lambda_t) F_t, \tag{10}$$

$$\text{where} \quad F_t \doteq i(S_t) + \gamma_t \sum_{k=0}^{t-1} \rho_k M_k \prod_{i=k+1}^{t-1} \gamma_i \lambda_i \rho_i = i(S_t) + \gamma_t \rho_{t-1} F_{t-1}, \quad \text{with } F_{-1} \doteq 0, \tag{11}$$

giving the final update for ETD(λ), derived from the forward-view update (9):

$$\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha \left(R_{t+1} + \gamma_{t+1} \boldsymbol{\theta}_t^\top \boldsymbol{\phi}_{t+1} - \boldsymbol{\theta}_t^\top \boldsymbol{\phi}_t \right) \mathbf{e}_t \tag{12}$$

$$\mathbf{e}_t \doteq \rho_t \left(\gamma_t \lambda_t \mathbf{e}_{t-1} + M_t \boldsymbol{\phi}_t \right), \quad \text{with } \mathbf{e}_{-1} \doteq \mathbf{0}. \tag{13}$$

The trace F_t here is similar to that of emphatic TD(0), adapted to the off-policy case through the application of ρ_t. According to (10), the emphasis M_t can be written simply as a linear interpolation between $i(S_t)$ and F_t. The per-step computational and memory complexity of ETD(λ) is the same as that of original TD(λ): $O(n)$ in the number of features. The additional cost ETD(λ) incurs due to the computation of the scalar emphasis is negligible.

4. Stability and convergence of ETD(λ)

We have discussed the motivations and ideas that led to the design of the emphasis weighting scheme (10)-(13) for ETD(λ). We now discuss several salient analytical properties underlying the algorithm due to this weighting scheme, and present the key stability and convergence results we have obtained for the algorithm. First, we formally state the conditions needed for the analysis.

Assumption 1 (Conditions on the target and behavior policies)

(i) The target policy π is such that $(\mathbf{I} - \mathbf{P}_\pi \mathbf{\Gamma})^{-1}$ exists, where $\mathbf{\Gamma}$ is the $N \times N$ diagonal matrix with the state-dependent discount factors $\gamma(s), s \in \mathcal{S}$, as its diagonal entries.

(ii) The behavior policy μ induces an irreducible Markov chain on $\mathcal{S}$, with the unique invariant distribution $d_\mu(s), s \in \mathcal{S}$, and for all $(s,a) \in \mathcal{S} \times \mathcal{A}$, $\mu(a|s) > 0$ if $\pi(a|s) > 0$.

Under Assumption 1(i), the value function v_π is specified by the expected total (discounted) rewards as $\boldsymbol{v}_\pi = (\mathbf{I} - \mathbf{P}_\pi \mathbf{\Gamma})^{-1} \mathbf{r}_\pi$; i.e., $\boldsymbol{v}_\pi$ is the unique solution of the Bellman equation $\boldsymbol{v} = \mathbf{r}_\pi + \mathbf{P}_\pi \mathbf{\Gamma} \boldsymbol{v}$. Associated with ETD($\lambda$) is a multistep, generalized Bellman equation which is determined by the bootstrapping parameters $\lambda(s)$ and also has $\boldsymbol{v}_\pi$ as its unique solution (Sutton 1995):

$$\boldsymbol{v} = \mathbf{r}_\pi^\lambda + \mathbf{P}_\pi^\lambda \boldsymbol{v}, \tag{14}$$

where $\mathbf{P}_\pi^\lambda$ is a substochastic matrix and $\mathbf{r}_\pi^\lambda \in \mathbb{R}^N$.[2] Let $\mathbf{\Phi}$ be the $N \times n$ matrix with the feature vectors $\boldsymbol{\phi}(s)^\top, s \in \mathcal{S}$, as its rows. The goal of ETD(λ) is to find an approximate solution of the Bellman equation (14) in the space $\{\mathbf{\Phi}\boldsymbol{\theta} \mid \boldsymbol{\theta} \in \mathbb{R}^n\}$.

Let us call those states on which ETD(λ) places positive emphasis weights *emphasized states*. More precisely, under Assumption 1(ii), we can assign an expected emphasis weight $m(s)$ for each state s, according to the weighting scheme (10)-(13), as (Sutton et al. 2015):

$$\big[\, m(1), m(2), \ldots, m(N) \,\big] = \mathbf{d}_{\mu,i}{}^\top (\mathbf{I} - \mathbf{P}_\pi^\lambda)^{-1}, \tag{15}$$

where $\mathbf{d}_{\mu,i} \in \mathbb{R}^N$ denotes the vector with components $d_{\mu,i}(s) = d_\mu(s) \cdot i(s), s \in \mathcal{S}$. Emphasized states are precisely those with $m(s) > 0$. It is important to observe from (15) that the emphasis weights $m(s)$ reflect the occupancy probabilities of the *target policy*, with respect to $\mathbf{P}_\pi^\lambda$ and an initial distribution proportional to $\mathbf{d}_{\mu,i}$, rather than the behavior policy. As will be seen shortly, this gives ETD(λ) a desired stability property that lacks normally in TD(λ) algorithms with selective updating.

Let $\mathbf{M}$ denote the diagonal matrix with the emphasis weights $m(s)$ on its diagonal. By considering the stationary case, the equation that ETD(λ) aims to solve is shown by Sutton et al. (2015) to be:

$$\mathbf{A}\boldsymbol{\theta} = \mathbf{b}, \quad \boldsymbol{\theta} \in \mathbb{R}^n, \tag{16}$$

$$\text{where} \qquad \mathbf{A} = \mathbf{\Phi}^\top \mathbf{M}\, (\mathbf{I} - \mathbf{P}_\pi^\lambda)\, \mathbf{\Phi}, \qquad \mathbf{b} = \mathbf{\Phi}^\top \mathbf{M}\, \mathbf{r}_\pi^\lambda. \tag{17}$$

In terms of the approximate value function $\boldsymbol{v} = \mathbf{\Phi}\boldsymbol{\theta}$, under a mild condition on the approximation architecture given below, the equation (16) is equivalent to a projected version of the Bellman equation (14):

$$\boldsymbol{v} = \Pi\big(\mathbf{r}_\pi^\lambda + \mathbf{P}_\pi^\lambda \boldsymbol{v}\big), \quad \boldsymbol{v} \in \big\{\mathbf{\Phi}\boldsymbol{\theta} \mid \boldsymbol{\theta} \in \mathbb{R}^n\big\}, \tag{18}$$

2. Specifically, with $\mathbf{\Lambda}$ denoting the diagonal matrix with $\lambda(s), s \in \mathcal{S}$, as its diagonal entries, we have $\mathbf{P}_\pi^\lambda = \mathbf{I} - (\mathbf{I} - \mathbf{P}_\pi \mathbf{\Gamma} \mathbf{\Lambda})^{-1} (\mathbf{I} - \mathbf{P}_\pi \mathbf{\Gamma})$ and $\mathbf{r}_\pi^\lambda = (\mathbf{I} - \mathbf{P}_\pi \mathbf{\Gamma} \mathbf{\Lambda})^{-1} \mathbf{r}_\pi$.

where Π denotes projection onto the approximation subspace with respect to a weighted Euclidean norm or seminorm $\|\cdot\|_m^2$, defined by the emphasis weights as $\|\boldsymbol{v}\|_m^2 = \sum_{s \in \mathcal{S}} m(s) v(s)^2$.

Assumption 2 (Condition on the approximation architecture)
The set of feature vectors of emphasized states, $\{\boldsymbol{\phi}(s) \mid s \in \mathcal{S},\, m(s) > 0\}$, contains n linearly independent vectors.

We note that Assumption 2 (which implies the linear independence of the columns of Φ) is satisfied in particular if the set of feature vectors, $\{\boldsymbol{\phi}(s) \mid s \in \mathcal{S},\, i(s) > 0\}$, contains n linearly independent vectors, since states with positive interest $i(s)$ are among the emphasized states. So this assumption can be easily satisfied in reinforcement learning without model knowledge.

We are now ready to discuss an important stability property underlying our algorithm. By making the emphasis weights $m(s)$ reflecting the occupancy probabilities of the target policy, as discussed earlier, the weighting scheme (10)-(13) of our algorithm ensures that the matrix $\mathbf{A}$ is positive definite under almost minimal conditions for off-policy training:[3]

Theorem 1 (Stability property of A) *Under Assumptions 1-2, the matrix $\mathbf{A}$ is positive definite (that is, there exists $c > 0$ such that $\boldsymbol{\theta}^\top \mathbf{A} \boldsymbol{\theta} \geq c \, \|\boldsymbol{\theta}\|_2^2$ for all $\boldsymbol{\theta} \in \mathbb{R}^n$).*

This property of $\mathbf{A}$ shows that the equation (16) associated with ETD(λ) has a unique solution $\boldsymbol{\theta}^*$ (equivalently, the equation (18) has the approximate value function $\boldsymbol{v} = \boldsymbol{\Phi}\boldsymbol{\theta}^*$ as its unique solution). Moreover, it shows that unlike normal TD(λ) with selective updating, here the deterministic update in the parameter space, $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha(\mathbf{A}\boldsymbol{\theta}_t - \mathbf{b})$, converges to $\boldsymbol{\theta}^*$ for sufficiently small stepsize α, and when diminishing stepsizes $\{\alpha_t\}$ are used in ETD(λ), $\{\boldsymbol{\theta}^*\}$ is globally asymptotically stable for the associated "mean ODE" $\dot{\boldsymbol{\theta}} = -\mathbf{A}\boldsymbol{\theta} + \mathbf{b}$ (Kushner & Yin 2003).[4] We are now ready to address the convergence of the algorithm.

Assumption 3 (Conditions on noisy rewards and diminishing stepsizes)

(i) The variances of the random rewards $\{R_t\}$ are bounded.

(ii) The (deterministic) stepsizes $\{\alpha_t\}$ satisfy that $\alpha_t = O(1/t)$ and $\frac{\alpha_t - \alpha_{t+1}}{a_t} = O(1/t)$.

Under the preceding assumptions, we have the following result, proved in (Yu 2015):[5]

3. The conclusion of Theorem 1 for the case of an interest function $i(\cdot) > 0$ is first proved by Sutton, Mahmood, and White (see their Theorem 1); Theorem 1 as given here is proved by Yu (2015) (see Prop. C.2 and Remark C.2 in Appendix C therein). The analyses in both works are motivated by a proof idea of Sutton (1988), which is to analyze the structure of the $N \times N$ matrix $\mathbf{M}(\mathbf{I} - \mathbf{P}_\pi^\lambda)$ and to invoke a result from matrix theory on strictly or irreducibly diagonally dominant matrices (Varga 2000, Cor. 1.22).
4. The important analytical properties discussed here can be shown to also extend to the case where the linear independence condition in Assumption 2 is relaxed: there, $\mathbf{A}$ acts like a positive definite matrix on the subspace of $\boldsymbol{\theta}$ (the range space of $\mathbf{A}$) that ETD(λ) naturally operates on. These extensions are based on both our understanding of how the weighting scheme (10)-(13) is designed (Sutton et al. 2015) and the special structure of the matrix $\mathbf{M}(\mathbf{I} - \mathbf{P}_\pi^\lambda)$ revealed in the proof of (Yu 2015, Prop. C.2). We will report the details of these extensions in a separate paper, however.
5. The proof is similar to but more complex than the convergence proof for off-policy LSTD/TD (Yu 2012). Among others, we show that despite the high variance in off-policy learning, the Markov chain $\{(S_t, A_t, \mathbf{e}_t, F_t)\}$ on the joint space $\mathcal{S} \times \mathcal{A} \times \mathbb{R}^{n+1}$ exhibits nice properties including ergodicity. We use these properties together with convergence results for a least-squares version of ETD(λ) and a convergence theorem from stochastic approximation theory (Kushner & Yin 2003, Theorem 6.1.1) to establish the desired convergence of ETD(λ) and its constrained variant by a "mean ODE" based proof method.

Theorem 2 (Convergence of ETD(λ))
Let Assumptions 1-3 hold. Then, for each initial $\boldsymbol{\theta}_0 \in \mathbb{R}^n$, the sequence $\{\boldsymbol{\theta}_t\}$ generated by ETD(λ) converges to $\boldsymbol{\theta}^*$ with probability 1.

To satisfy the stepsize Assumption 3(ii), we can take $\alpha_t = c_1/(c_2+t)$ for some constants $c_1, c_2 > 0$, for example. If the behavior policy is close to the target policy, we believe that ETD(λ) also converges for larger stepsizes.

5. An illustrative experiment

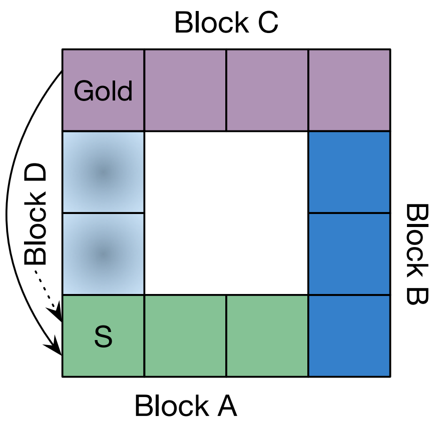


Figure 1. The *Miner* problem where a miner continually collects gold from the `Gold` cell until it falls into a trap, which can be activated in `Block D`.

In this section we describe an experiment to illustrate the flexibility and benefits of ETD(λ) in learning several off-policy predictions in terms of value estimates.

In this experiment we used a gridworld problem depicted in Figure 1, which we call the *Miner* problem. Here a miner starting from the cell `S` continually wandered around the gridworld using one of the following actions: `left`, `right`, `up` and `down`, each indicating the direction of the miner's movement. An invalid direction such as going down from `S` resulted in no movement. The miner got zero reward at every transition except when it arrived at the cell denoted by `Gold`, in which case a +1 reward was obtained. There were two routes to reach the `Gold` cell from `S`: one went straight up through `Block D`, and the other was roundabout through `Block B`. A trap could be activated in one of the two cells in `Block D` chosen randomly. Once active, a trap stayed for 3 time steps, and only one trap was active at any time. The trap activation probability was 0.25. If the miner arrived at the `Gold` cell or fell into a trap, it was transported to `S` in the next time step. Note that arriving at the `Gold` cell or a trap was not the end of an episode, and the miner wandered around continually.

The miner followed a fixed behavior policy according to which the miner was equally likely to take any of the four actions in `Block A`, more inclined to go up in both `Block B` and `Block D`, and more inclined to go left in `Block C`, in each case with probability 0.4. The rest of the actions were equally likely.

We evaluated three fixed policies different than the behavior policy. We call them `uniform`, `headfirst` and `cautious` policies. Under the `uniform` policy, all actions were equally likely in every cell. Under the `headfirst` policy, the miner chose to go up in `Block A` and `D` with 0.9 probability while other actions from those blocks were equally likely. All the actions from other blocks were chosen with equal probability. Under the `cautious` policy, the miner was more inclined to go right in `Block A`, go up in both `Block B` and `Block D`, and go left in `Block C`, in each case with probability 0.6. The rest of the actions were equally likely.

We were interested to predict how much gold the miner could collect before falling into a trap if the miner had used the above three policies, without executing any of these policies.

We set $\gamma = 0$ for those states where the miner got entrapped to indicate termination under the target policy (although behavior policy continued) and a discounting of $\gamma = 0.99$ in other states. We set $i(s) = 1$ whenever the miner was in `Block A` and 0 everywhere else. As the behavior policy of the miner is different than the three target policies, it must use off-policy training to learn what could happen under each of those policies. We used three instances of ETD(λ) for three different predictions, each using $\alpha = 0.001$, $\lambda = 1.0$ when the miner was in `Block D`, $\lambda = 0$ in `Block A`, and $\lambda = 0.9$ in other states. We clipped each component of the increment to $\boldsymbol{\theta}$ in (12) between -0.5 and $+0.5$ in order to reduce the impact of extremely large eligibility traces on updates. Clipping the increments can be shown to be theoretically sound, although we will not discuss this subject here. The state representation used four features: each corresponding to the miner being in one of the four blocks. The miner wandered continually until 3000 entrapments occurred.

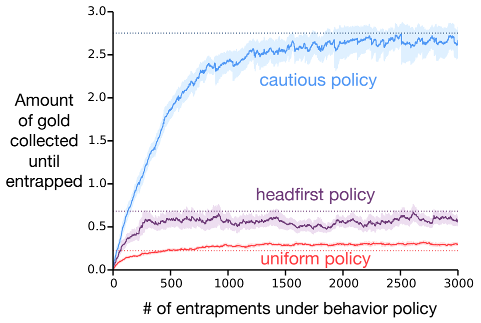


Figure 2. Simultaneous evaluation of three policies different than the behavior policy using ETD(λ).

Figure 2 shows estimates calculated by ETD(λ) in terms of its weight corresponding to `Block A` for the three target policies. The curves shown are average estimates with two standard error bands using 50 independent runs. The dotted straight lines indicate the true state value estimated through Monte Carlo simulation from `S`. Due to the use of function approximation and clipping of the updates, the true value could not be estimated accurately. However, the estimates for the three policies appear to approach values close to the true ones, and they preserved the relative ordering of the policies. In the absence of the clipping, the estimates were less stable and highly volatile, occasionally moving far away from the desired value for some of the runs. Although some of the learning curves still look volatile, clipping the updates reduced its extent considerably.

6. Discussion and conclusions

We summarized the motivations, key ideas and the available results on emphatic algorithms. Furthermore, we demonstrated how ETD(λ) can be used to learn many predictions about the world simultaneously using off-policy learning, and the flexibility it provides through state-dependent discounting, bootstrapping and user-specified relative interests to states. ETD(λ) is among the few algorithms with per-step linear computational complexity that are convergent under off-policy training. Compared to convergent gradient-based TD algorithms (Maei 2011), ETD(λ) is simpler and easier to use; it has only one learned parameter vector and one step-size parameter. The problem of high variance is common in off-policy learning, and ETD(λ) is susceptible to it as well. An extension to variance-reduction methods, such as weighted importance sampling (Precup et al. 2000, Mahmood et al. 2014, 2015), can be a natural remedy to this problem. ETD(λ) produces a different algorithm than the conventional TD(λ) even in the on-policy case. It is likely that, in many cases, ETD(λ) provides more accurate predictions than TD(λ) through the use of relative interests and emphasis. An interesting direction for future work would be to characterize these cases.

References

Baird, L. C. (1995). Residual algorithms: Reinforcement learning with function approximation. In *Proceedings of the 12th International Conference on Machine Learning*, pp. 30–37.

Bertsekas, D. P., Tsitsiklis, J. N. (1996). *Neuro-Dynamic Programming*. Athena Scientific.

Boyan, J. A., (1999). Least-squares temporal difference learning. In *Proceedings of the 16th International Conference on Machine Learning*, pp. 49–56.

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

Kushner, H. J., Yin G. G. (2003). *Stochastic Approximation and Recursive Algorithms and Applications*, second edition. Springer-Verlag.

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

Maei, H. R. (2011). *Gradient Temporal-Difference Learning Algorithms*. PhD thesis, University of Alberta.

Mahmood, A. R., van Hasselt, H., Sutton, R. S. (2014). Weighted importance sampling for off-policy learning with linear function approximation. *Advances in Neural Information Processing Systems 27*.

Mahmood, A. R., Sutton, R. S. (2015). Off-policy learning based on weighted importance sampling with linear computational complexity. In *Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence*, Amsterdam, Netherlands.

Precup, D., Sutton, R. S., Singh, S. (2000). Eligibility traces for off-policy policy evaluation. In *Proceedings of the 17th International Conference on Machine Learning*, pp. 759–766. Morgan Kaufmann.

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

Sutton R. S. (1995). TD models: Modeling the world at a mixture of time scales. In *Proceedings of the 12th International Conference on Machine Learning*.

Sutton, R. S., Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press.

Sutton, R. S., Mahmood, A. R., White, M. (2015). An emphatic approach to the problem of off-policy temporal-difference learning, arXiv:1503.04269 [cs.LG].

Thomas, P. (2014). Bias in natural actor–critic algorithms. In *Proceedings of the 31st International Conference on Machine Learning*. JMLR W&CP 32(1):441–448.

Tsitsiklis, J. N., Van Roy, B. (1996). Feature-based methods for large scale dynamic programming. *Machine Learning* 22:59–94.

Varga, R. S. (2000). *Matrix Iterative Analysis*, second edition. Springer-Verlag.

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

van Seijen, H., & Sutton, R. S. (2014). True online TD(λ). In *Proceedings of the 31st International Conference on Machine Learning*. JMLR W&CP 32(1):692–700.

Yu, H. (2012). Least squares temporal difference methods: An analysis under general conditions. *SIAM Journal on Control and Optimization* 50:3310-3343.

Yu, H. (2015). On convergence of emphatic temporal-difference learning. In *Proceedings of the 28th Annual Conference on Learning Theory*. Paris, France.