Submitted 11/14; Revised 11/15; Published 5/16

# An Emphatic Approach to the Problem of Off-policy Temporal-Difference Learning

**Richard S. Sutton** SUTTON@CS.UALBERTA.CA
**A. Rupam Mahmood** ASHIQUE@CS.UALBERTA.CA
**Martha White** WHITEM@CS.UALBERTA.CA
*Reinforcement Learning and Artificial Intelligence Laboratory*
*Department of Computing Science, University of Alberta*
*Edmonton, Alberta, Canada T6G 2E8*

**Editor:** Shie Mannor

## Abstract

In this paper we introduce the idea of improving the performance of parametric temporal-difference (TD) learning algorithms by selectively emphasizing or de-emphasizing their updates on different time steps. In particular, we show that varying the emphasis of linear TD($\lambda$)'s updates in a particular way causes its expected update to become stable under off-policy training. The only prior model-free TD methods to achieve this with per-step computation linear in the number of function approximation parameters are the gradient-TD family of methods including TDC, GTD($\lambda$), and GQ($\lambda$). Compared to these methods, our *emphatic TD($\lambda$)* is simpler and easier to use; it has only one learned parameter vector and one step-size parameter. Our treatment includes general state-dependent discounting and bootstrapping functions, and a way of specifying varying degrees of interest in accurately valuing different states.

**Keywords:** Temporal-difference learning, Off-policy learning, Function approximation, Stability, Convergence.

## 1. Parametric Temporal-Difference Learning

Temporal-difference (TD) learning is perhaps the most important idea to come out of the field of reinforcement learning. The problem it solves is that of efficiently learning to make a sequence of long-term predictions about how a dynamical system will evolve over time. The key idea is to use the change (temporal difference) from one prediction to the next as an error in the earlier prediction. For example, if you are predicting on each day what the stock-market index will be at the end of the year, and events lead you one day to make a much lower prediction, then a TD method would infer that the predictions made prior to the drop were probably too high; it would adjust the parameters of its prediction function so as to make lower predictions for similar situations in the future. This approach contrasts with conventional approaches to prediction, which wait until the end of the year when the final stock-market index is known before adjusting any parameters, or else make only short-term (e.g., one-day) predictions and then iterate them to produce a year-end prediction. The TD approach is more convenient computationally because it requires less memory and because its computations are spread out uniformly over the year (rather than being bunched together all at the end of the year). A less obvious advantage of the TD approach is that it often produces statistically more accurate answers than conventional approaches (Sutton 1988).

Parametric temporal-difference learning was first studied as the key "learning by generalization" algorithm in Samuel's (1959) checker player. Sutton (1988) introduced the TD($\lambda$) algorithm and proved convergence in the mean of episodic linear TD(0), the simplest parametric TD method. The potential power of parametric TD learning was convincingly demonstrated by Tesauro (1992, 1995) when he applied TD($\lambda$) combined with neural networks and self play to obtain ultimately the world's best backgammon player. Dayan (1992) proved convergence in expected value of episodic linear TD($\lambda$) for all $\lambda \in [0, 1]$, and Tsitsiklis and Van Roy (1997) proved convergence with probability one of discounted continuing linear TD($\lambda$). Watkins (1989) extended TD learning to control in the form of Q-learning and proved its convergence in the tabular case (without function approximation, Watkins & Dayan 1992), while Rummery (1995) extended TD learning to control in an on-policy form as the Sarsa($\lambda$) algorithm. Bradtke and Barto (1996), Boyan (1999), and Nedic and Bertsekas (2003) extended linear TD learning to a least-squares form called LSTD($\lambda$). Parametric TD methods have also been developed as models of animal learning (e.g., Sutton & Barto 1990, Klopf 1988, Ludvig, Sutton & Kehoe 2012) and as models of the brain's reward systems (Schultz, Dayan & Montague 1997), where they have been particularly influential (e.g., Niv & Schoenbaum 2008, O'Doherty 2012). Sutton (2009, 2012) has suggested that parametric TD methods could be key not just to learning about reward, but to the learning of world knowledge generally, and to perceptual learning. Extensive analysis of parametric TD learning as stochastic approximation is provided by Bertsekas (2012, Chapter 6) and Bertsekas and Tsitsiklis (1996).

Within reinforcement learning, TD learning is typically used to learn approximations to the value function of a Markov decision process (MDP). Here the value of a state $s$, denoted $v_\pi(s)$, is defined as the sum of the expected long-term discounted rewards that will be received if the process starts in $s$ and subsequently takes actions as specified by the decision-making policy $\pi$, called the *target policy*. If there are a small number of states, then it may be practical to approximate the function $v_\pi$ by a table, but more generally a parametric form is used, such as a polynomial, multi-layer neural network, or linear mapping. Also key is the source of the data, in particular, the policy used to interact with the MDP. If the data is obtained while following the target policy $\pi$, then good convergence results are available for linear function approximation. This case is called *on-policy* learning because learning occurs while "on" the policy being learned about. In the alternative, *off-policy* case, one seeks to learn about $v_\pi$ while behaving (selecting actions) according to a different policy called the *behavior policy*, which we denote by $\mu$. Baird (1995) showed definitively that parametric TD learning was much less robust in the off-policy case (for $\lambda < 1$) by exhibiting counterexamples for which both linear TD(0) and linear Q-learning had unstable expected updates and, as a result, the parameters of their linear function approximation diverged to infinity. This is a serious limitation, as the off-policy aspect is key to Q-learning (perhaps the single most popular reinforcement learning algorithm), to learning from historical data and from demonstrations, and to the idea of using TD learning for perception and world knowledge.

Over the years, several different approaches have been taken to solving the problem of off-policy learning with TD learning ($\lambda < 1$). Baird (1995) proposed an approach based on gradient descent in the Bellman error for general parametric function approximation that has the desired computational properties, but which requires access to the MDP for double sampling and which in practice often learns slowly. Gordon (1995, 1996) proposed restricting attention to function approximators that are averagers, but this does not seem to be possible without storing many of the training examples, which would defeat the primary strength that we seek to obtain from parametric function approximation. The LSTD($\lambda$) method was always relatively robust to off-policy training (e.g., Lagoudakis & Parr 2003, Yu 2010, Mahmood, van Hasselt & Sutton 2014), but its per-step computational complexity is quadratic in the number of parameters of the function approximator, as opposed to the linear complexity of TD($\lambda$) and the other methods. Perhaps the most successful approach to date is the gradient-TD approach (e.g., Maei 2011, Sutton et al. 2009, Maei et al. 2010), including hybrid methods such as HTD (Hackman 2012). Gradient-TD methods are of linear complexity and guaranteed to converge for appropriately chosen step-size parameters but are more complex than TD($\lambda$) because they require a second auxiliary set of parameters with a second step size that must be set in a problem-dependent way for good performance. The studies by White (2015), Geist and Scherrer (2014), and Dann, Neumann, and Peters (2014) are the most extensive empirical explorations of gradient-TD and related methods to date.

In this paper we explore a new approach to solving the problem of off-policy TD learning with function approximation. The approach has novel elements but is similar to that developed by Precup, Sutton, and Dasgupta in 2001. They proposed to use importance sampling to reweight the updates of linear TD($\lambda$), emphasizing or de-emphasizing states as they were encountered, and thereby create a weighting equivalent to the stationary distribution under the target policy, from which the results of Tsitsiklis and Van Roy (1997) would apply and guarantee convergence. As we discuss later, this approach has very high variance and was eventually abandoned in favor of the gradient-TD approach. The new approach we explore in this paper is similar in that it also varies emphasis so as to reweight the distribution of linear TD($\lambda$) updates, but to a different goal. The new goal is to create a weighting equivalent to the *followon distribution* for the target policy started in the stationary distribution of the behavior policy. The followon distribution weights states according to how often they would occur prior to termination by discounting *if the target policy was followed*.

Our main result is to prove that varying emphasis according to the followon distribution produces a new version of linear TD($\lambda$), *called emphatic TD($\lambda$)*, that is stable under general off-policy training. By "stable" we mean that the expected update over the ergodic distribution (Tsitsiklis & Van Roy 1997) is a contraction, involving a positive definite matrix. We concentrate on stability in this paper because it is a prerequisite for full convergence of the stochastic algorithm. Demonstrations that the linear TD($\lambda$) is not stable under off-policy training have been the focus of previous counterexamples (Baird 1995, Tsitsiklis & Van Roy 1996, 1997, see Sutton & Barto 1998). Substantial additional theoretical machinery would be required for a full convergence proof. Recent work by Yu (2015a,b) builds on our stability result to prove that the emphatic TD($\lambda$) converges with probability one.

In this paper we first treat the simplest algorithm for which the difficulties of off-policy temporal-difference (TD) learning arise—the TD(0) algorithm with linear function approximation. We examine the conditions under which the expected update of on-policy TD(0) is stable, then why those conditions do not apply under off-policy training, and finally how they can be recovered for off-policy training using established importance-sampling methods together with the emphasis idea. After introducing the basic idea of emphatic algorithms using the special case of TD(0), we then develop the general case. In particular, we consider a case with general state-dependent discounting and bootstrapping functions, and with a user-specified allocation of function approximation resources. Our new theoretical results and the emphatic TD($\lambda$) algorithm are presented fully for this general case. Empirical examples elucidating the main theoretical results are presented in the last section prior to the conclusion.

## 2. On-policy Stability of TD(0)

To begin, let us review the conditions for stability of conventional TD($\lambda$) under on-policy training with data from a continuing finite Markov decision process. Consider the simplest function approximation case, that of linear TD($\lambda$) with $\lambda = 0$ and constant discount-rate parameter $\gamma \in [0, 1)$. Conventional linear TD(0) is defined by the following update to the parameter vector $\boldsymbol{\theta}_t \in \mathbb{R}^n$, made at each of a sequence of time steps $t = 0, 1, 2, \ldots$, on transition from state $S_t \in \mathcal{S}$ to state $S_{t+1} \in \mathcal{S}$, taking action $A_t \in \mathcal{A}$ and receiving reward $R_{t+1} \in \mathbb{R}$:

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

where $\alpha > 0$ is a step-size parameter, and $\boldsymbol{\phi}(s) \in \mathbb{R}^n$ is the feature vector corresponding to state $s$. The notation "$\doteq$" indicates an equality by definition rather than one that follows from previous definitions. In on-policy training, the actions are chosen according to a target policy $\pi : \mathcal{A} \times \mathcal{S} \to [0, 1]$, where $\pi(a|s) \doteq \mathbb{P}\{A_t = a | S_t = s\}$. The state and action sets $\mathcal{S}$ and $\mathcal{A}$ are assumed to be finite, but the number of states is assumed much larger than the number of learned parameters, $|\mathcal{S}| \doteq N \gg n$, so that function approximation is necessary. We use linear function approximation, in which the inner product of the parameter vector and the feature vector for a state is meant to be an approximation to the value of that state:

$$\boldsymbol{\theta}_t^\top \boldsymbol{\phi}(s) \approx v_\pi(s) \doteq \mathbb{E}_\pi[G_t | S_t = s], \tag{2}$$

where $\mathbb{E}_\pi[\cdot]$ denotes an expectation conditional on all actions being selected according to $\pi$, and $G_t$, the *return* at time $t$, is defined by

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

The TD(0) update (1) can be rewritten to make the stability issues more transparent:

$$\begin{aligned} \boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t + \alpha \Big( \underbrace{R_{t+1}\boldsymbol{\phi}(S_t)}_{\mathbf{b}_t \in \mathbb{R}^n} - \underbrace{\boldsymbol{\phi}(S_t) \left(\boldsymbol{\phi}(S_t) - \gamma \boldsymbol{\phi}(S_{t+1})\right)^\top}_{\mathbf{A}_t \in \mathbb{R}^{n \times n}} \boldsymbol{\theta}_t \Big) \\ &= \boldsymbol{\theta}_t + \alpha(\mathbf{b}_t - \mathbf{A}_t \boldsymbol{\theta}_t) \\ &= (\mathbf{I} - \alpha \mathbf{A}_t)\boldsymbol{\theta}_t + \alpha \mathbf{b}_t. \end{aligned} \tag{4}$$

The matrix $\mathbf{A}_t$ multiplies the parameter $\boldsymbol{\theta}_t$ and is thereby critical to the stability of the iteration. To develop intuition, consider the special case in which $\mathbf{A}_t$ is a diagonal matrix. If any of the diagonal elements are negative, then the corresponding diagonal element of $\mathbf{I} - \alpha\mathbf{A}_t$ will be greater than one, and the corresponding component of $\boldsymbol{\theta}_t$ will be amplified, which will lead to divergence if continued. (The second term ($\alpha\mathbf{b}_t$) does not affect the stability of the iteration.) On the other hand, if the diagonal elements of $\mathbf{A}_t$ are all positive, then $\alpha$ can be chosen smaller than one over the largest of them, such that $\mathbf{I}-\alpha\mathbf{A}_t$ is diagonal with all diagonal elements between 0 and 1. In this case the first term of the update tends to shrink $\boldsymbol{\theta}_t$, and stability is assured. In general, $\boldsymbol{\theta}_t$ will be reduced toward zero whenever $\mathbf{A}_t$ is positive definite.[1]

In actuality, however, $\mathbf{A}_t$ and $\mathbf{b}_t$ are random variables that vary from step to step, in which case stability is determined by the steady-state expectation, $\lim_{t\to\infty} \mathbb{E}[\mathbf{A}_t]$. In our setting, after an initial transient, states will be visited according to the steady-state distribution under $\pi$ (which we assume exists). We represent this distribution by a vector $\mathbf{d}_\pi$, each component of which gives the limiting probability of being in a particular state[2] $[\mathbf{d}_\pi]_s \doteq d_\pi(s) \doteq \lim_{t\to\infty} \mathbb{P}\{S_t\!=\!s\}$, which we assume exists and is positive at all states (any states not visited with nonzero probability can be removed from the problem). The special property of the steady-state distribution is that once the process is in it, it remains in it. Let $\mathbf{P}_\pi$ denote the $N \times N$ matrix of transition probabilities $[\mathbf{P}_\pi]_{ij} \doteq \sum_a \pi(a|i)p(j|i,a)$ where $p(j|i,a) \doteq \mathbb{P}\{S_{t+1}\!=\!j|S_t\!=\!i, A_t\!=\!a\}$. Then the special property of $\mathbf{d}_\pi$ is that

$$\mathbf{P}_\pi^\top \mathbf{d}_\pi = \mathbf{d}_\pi. \tag{5}$$

Consider any stochastic algorithm of the form (4), and let $\mathbf{A} \doteq \lim_{t\to\infty} \mathbb{E}[\mathbf{A}_t]$ and $\mathbf{b} \doteq \lim_{t\to\infty} \mathbb{E}[\mathbf{b}_t]$. We define the stochastic algorithm to be *stable* if and only if the corresponding deterministic algorithm,

$$\bar{\boldsymbol{\theta}}_{t+1} \doteq \bar{\boldsymbol{\theta}}_t + \alpha(\mathbf{b} - \mathbf{A}\bar{\boldsymbol{\theta}}_t), \tag{6}$$

is convergent to a unique fixed point independent of the initial $\bar{\boldsymbol{\theta}}_0$. This will occur iff the $\mathbf{A}$ matrix has a full set of eigenvalues all of whose real parts are positive. If a stochastic algorithm is stable and $\alpha$ is reduced according to an appropriate schedule, then its parameter vector may converge with probability one. However, in this paper we focus only on stability as a prerequisite for convergence (of the original stochastic algorithm), leaving convergence itself to future work. If the stochastic algorithm converges, it is to a fixed point $\bar{\boldsymbol{\theta}}$ of the deterministic algorithm, at which $\mathbf{A}\bar{\boldsymbol{\theta}} = \mathbf{b}$, or $\bar{\boldsymbol{\theta}} = \mathbf{A}^{-1}\mathbf{b}$. (Stability assures existence of the inverse.) In this paper we focus on establishing stability by proving that $\mathbf{A}$ is positive definite. From definiteness it immediately follows that $\mathbf{A}$ has a full set of eigenvectors (because $\mathbf{y}^\top\mathbf{A}\mathbf{y} > 0, \forall \mathbf{y} \neq \mathbf{0}$) and that the corresponding eigenvalues all have real parts.[3]

[1] A real matrix $\mathbf{A}$ is defined to be *positive definite* in this paper iff $\mathbf{y}^\top\mathbf{A}\mathbf{y} > 0$ for any real vector $\mathbf{y} \neq \mathbf{0}$.

[2] Here and throughout the paper we use brackets with subscripts to denote the individual elements of vectors and matrices.

[3] To see the latter, let $\mathrm{Re}(x)$ denote the real part of a complex number $x$, and let $\mathbf{y}^*$ denotes the conjugate transpose of a complex vector $\mathbf{y}$. Then, for any eigenvalue–eigenvector pair $\lambda, \mathbf{y}$: $0 < \mathrm{Re}(\mathbf{y}^*\mathbf{A}\mathbf{y}) = \mathrm{Re}(\mathbf{y}^*\lambda\mathbf{y}) = \mathrm{Re}(\lambda)\mathbf{y}^*\mathbf{y} \implies 0 < \mathrm{Re}(\lambda)$.

Now let us return to analyzing on-policy TD(0). Its $\mathbf{A}$ matrix is

$$\begin{aligned}\mathbf{A} = \lim_{t\to\infty} \mathbb{E}[\mathbf{A}_t] &= \lim_{t\to\infty} \mathbb{E}_\pi\Big[\boldsymbol{\phi}(S_t)\left(\boldsymbol{\phi}(S_t) - \gamma\boldsymbol{\phi}(S_{t+1})\right)^\top\Big] \\ &= \sum_s d_\pi(s)\,\boldsymbol{\phi}(s)\left(\boldsymbol{\phi}(s) - \gamma\sum_{s'}[\mathbf{P}_\pi]_{ss'}\boldsymbol{\phi}(s')\right)^\top \\ &= \boldsymbol{\Phi}^\top\mathbf{D}_\pi(\mathbf{I} - \gamma\mathbf{P}_\pi)\boldsymbol{\Phi},\end{aligned}$$

where $\boldsymbol{\Phi}$ is the $N \times n$ matrix with the $\boldsymbol{\phi}(s)$ as its rows, and $\mathbf{D}_\pi$ is the $N \times N$ diagonal matrix with $\mathbf{d}_\pi$ on its diagonal. This $\mathbf{A}$ matrix is typical of those we consider in this paper in that it consists of $\boldsymbol{\Phi}^\top$ and $\boldsymbol{\Phi}$ wrapped around a distinctive $N \times N$ matrix that varies with the algorithm and the setting, and which we call the *key matrix*. An $\mathbf{A}$ matrix of this form will be positive definite whenever the corresponding key matrix is positive definite.[4] In this case the key matrix is $\mathbf{D}_\pi(\mathbf{I} - \gamma\mathbf{P}_\pi)$.

For a key matrix of this type, positive definiteness is assured if all of its columns sum to nonnegative numbers. This was shown by Sutton (1988, p. 27) based on two previously established theorems. One theorem says that any matrix $\mathbf{M}$ is positive definite if and only if the symmetric matrix $\mathbf{S} = \mathbf{M}+\mathbf{M}^\top$ is positive definite (Sutton 1988, appendix). The second theorem says that any symmetric real matrix $\mathbf{S}$ is positive definite if the absolute values of its diagonal entries are greater than the sum of the absolute values of the corresponding off-diagonal entries (Varga 1962, p. 23). For our key matrix, $\mathbf{D}_\pi(\mathbf{I} - \gamma\mathbf{P}_\pi)$, the diagonal entries are positive and the off-diagonal entries are negative, so all we have to show is that each row sum plus the corresponding column sum is positive. The row sums are all positive because $\mathbf{P}_\pi$ is a stochastic matrix and $\gamma < 1$. Thus it only remains to show that the column sums are nonnegative.

Note that the row vector of the column sums of any matrix $\mathbf{M}$ can be written as $\mathbf{1}^\top\mathbf{M}$, where $\mathbf{1}$ is the column vector with all components equal to 1. The column sums of our key matrix, then, are:

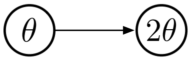
$$\begin{aligned}\mathbf{1}^\top\mathbf{D}_\pi(\mathbf{I} - \gamma\mathbf{P}_\pi) &= \mathbf{d}_\pi^\top(\mathbf{I} - \gamma\mathbf{P}_\pi) \\ &= \mathbf{d}_\pi^\top - \gamma\mathbf{d}_\pi^\top\mathbf{P}_\pi \\ &= \mathbf{d}_\pi^\top - \gamma\mathbf{d}_\pi^\top && \text{(by (5))} \\ &= (1-\gamma)\mathbf{d}_\pi,\end{aligned}$$

all components of which are positive. Thus, the key matrix and its $\mathbf{A}$ matrix are positive definite, and on-policy TD(0) is stable. Additional conditions and a schedule for reducing $\alpha$ over time (as in Tsitsiklis and Van Roy 1997) are needed to prove convergence with probability one, $\boldsymbol{\theta}_\infty = \mathbf{A}^{-1}\mathbf{b}$, but the analysis above includes the most important steps that vary from algorithm to algorithm.

[4] Strictly speaking, positive definiteness of the key matrix assures only that $\mathbf{A}$ is positive *semi*-definite, because it is possible that $\boldsymbol{\Phi}\mathbf{y} = \mathbf{0}$ for some $\mathbf{y} \neq \mathbf{0}$, in which case $\mathbf{y}^\top\mathbf{A}\mathbf{y}$ will be zero as well. To rule this out, we assume, as is commonly done, that the columns of $\boldsymbol{\Phi}$ are linearly independent (i.e., that the features are not redundant), and thus that $\boldsymbol{\Phi}\mathbf{y} = \mathbf{0}$ only if $\mathbf{y} = \mathbf{0}$. If this were not true, then convergence (if it occurs) may not be to a unique $\boldsymbol{\theta}_\infty$, but rather to a subspace of parameter vectors all of which produce the same approximate value function.

## 3. Instability of Off-policy TD(0)

Before developing the off-policy setting in detail, it is useful to understand informally why TD(0) is susceptible to instability. TD learning involves learning an estimate from an estimate, which can be problematic if there is generalization between the two estimates. For example, suppose there is a transition between two states with the same feature representation except that the second is twice as big:



where here $\theta$ and $2\theta$ are the estimated values of the two states—that is, their feature representations are a single feature that is 1 for the first state and 2 for the second (cf. Tsitsiklis & Van Roy 1996). Now suppose that $\theta$ is 10 and the reward on the transition is 0. The transition is then from a state valued at 10 to a state valued at 20. If $\gamma$ is near 1 and $\alpha$ is 0.1, then $\theta$ will be increased to approximately 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 $\theta$, to approximately 12.1. If this transition is experienced repeatedly on its own, then the system is unstable and the parameter increases without bound—it diverges. We call this the $\theta \to 2\theta$ problem.

In on-policy learning, repeatedly experiencing just this single problematic transition cannot happen, because, after the highly-valued $2\theta$ state has been entered, it must then be exited. The transition from it will either be to a lesser or equally-valued state, in which case $\theta$ will be significantly decreased, or to an even higher-valued state which in turn must be followed by an even larger decrease in its estimated value or a still higher-valued state. Eventually, the promise of high value must be made good in the form of a high reward, or else estimates will be decreased, and this ultimately constrains $\theta$ and forces stability and convergence. In the off-policy case, however, if there is a deviation from the target policy then the promise is excused and need never be fulfilled. Later in this section we present a complete example of how the $\theta \to 2\theta$ problem can cause instability and divergence under off-policy training.

With these intuitions, we now detail our off-policy setting. As in the on-policy case, the data is a single, infinite-length trajectory of actions, rewards, and feature vectors generated by a continuing finite Markov decision process. The difference is that the actions are selected not according to the target policy $\pi$, but according to a different *behavior policy* $\mu : \mathcal{A} \times \mathcal{S} \to [0, 1]$, yet still we seek to estimate state values under $\pi$ (as in (2)). Of course, it would be impossible to estimate the values under $\pi$ if the actions that $\pi$ would take were never taken by $\mu$ and their consequences were never observed. Thus we assume that $\mu(a|s) > 0$ for every state and action for which $\pi(a|s) > 0$. This is called the assumption of *coverage*. It is trivially satisfied by any $\epsilon$-greedy or soft behavior policy. As before we assume that there is a stationary distribution $d_\mu(s) \doteq \lim_{t\to\infty} \mathbb{P}\{S_t = s\} > 0, \forall s \in \mathcal{S}$, with corresponding $N$-vector $\mathbf{d}_\mu$.

Even if there is coverage, the behavior policy will choose actions with proportions different from the target policy. For example, some actions taken by $\mu$ might never be chosen by $\pi$. To address this, we use importance sampling to correct for the relative probability of taking the action actually taken, $A_t$, in the state actually encountered, $S_t$, under the target and behavior policies:

$$\rho_t \doteq \frac{\pi(A_t|S_t)}{\mu(A_t|S_t)}.$$

This quantity is called the *importance sampling ratio* at time $t$. Note that its expected value is one:

$$\mathbb{E}_\mu[\rho_t|S_t\!=\!s] = \sum_a \mu(a|s)\frac{\pi(a|s)}{\mu(a|s)} = \sum_a \pi(a|s) = 1.$$

The ratio will be exactly one only on time steps on which the action probabilities for the two policies are exactly the same; these time steps can be treated the same as in the on-policy case. On other time steps the ratio will be greater or less than one depending on whether the action taken was more or less likely under the target policy than under the behavior policy, and some kind of correction is needed.

In general, for any random variable $Z_{t+1}$ dependent on $S_t$, $A_t$ and $S_{t+1}$, we can recover its expectation under the target policy by multiplying by the importance sampling ratio:

$$\begin{aligned}\mathbb{E}_\mu[\rho_t Z_{t+1}|S_t\!=\!s] &= \sum_a \mu(a|s)\frac{\pi(a|s)}{\mu(a|s)}Z_{t+1}\\ &= \sum_a \pi(a|s)Z_{t+1}\\ &= \mathbb{E}_\pi[Z_{t+1}|S_t\!=\!s], \qquad \forall s\in\mathcal{S}. \end{aligned} \tag{7}$$

We can use this fact to begin to adapt TD(0) for off-policy learning (Precup, Sutton & Singh 2000). We simply multiply the whole TD(0) update (1) by $\rho_t$:

$$\begin{aligned}\boldsymbol{\theta}_{t+1} &\doteq \boldsymbol{\theta}_t + \rho_t\,\alpha\left(R_{t+1} + \gamma\boldsymbol{\theta}_t^\top\boldsymbol{\phi}_{t+1} - \boldsymbol{\theta}_t^\top\boldsymbol{\phi}_t\right)\boldsymbol{\phi}_t \qquad (8)\\ &= \boldsymbol{\theta}_t + \alpha\Big(\underbrace{\rho_t R_{t+1}\boldsymbol{\phi}_t}_{\mathbf{b}_t} - \underbrace{\rho_t\boldsymbol{\phi}_t\left(\boldsymbol{\phi}_t - \gamma\boldsymbol{\phi}_{t+1}\right)^\top}_{\mathbf{A}_t}\boldsymbol{\theta}_t\Big),\end{aligned}$$

where here we have used the shorthand $\boldsymbol{\phi}_t \doteq \boldsymbol{\phi}(S_t)$. Note that if the action taken at time $t$ is never taken under the target policy in that state, then $\rho_t = 0$ and there is no update on that step, as desired. We call this algorithm *off-policy TD(0)*.

Off-policy TD(0)'s $\mathbf{A}$ matrix is

$$\begin{aligned}\mathbf{A} = \lim_{t\to\infty}\mathbb{E}[\mathbf{A}_t] &= \lim_{t\to\infty}\mathbb{E}_\mu\left[\rho_t\boldsymbol{\phi}_t\left(\boldsymbol{\phi}_t - \gamma\boldsymbol{\phi}_{t+1}\right)^\top\right]\\ &= \sum_s d_\mu(s)\mathbb{E}_\mu\left[\rho_k\boldsymbol{\phi}_k\left(\boldsymbol{\phi}_k - \gamma\boldsymbol{\phi}_{k+1}\right)^\top\middle|S_k = s\right]\\ &= \sum_s d_\mu(s)\mathbb{E}_\pi\left[\boldsymbol{\phi}_k\left(\boldsymbol{\phi}_k - \gamma\boldsymbol{\phi}_{k+1}\right)^\top\middle|S_k = s\right] \qquad \text{(by (7))}\\ &= \sum_s d_\mu(s)\,\boldsymbol{\phi}(s)\left(\boldsymbol{\phi}(s) - \gamma\sum_{s'}[\mathbf{P}_\pi]_{ss'}\boldsymbol{\phi}(s')\right)^\top\\ &= \mathbf{\Phi}^\top\mathbf{D}_\mu(\mathbf{I} - \gamma\mathbf{P}_\pi)\mathbf{\Phi},\end{aligned}$$

where $\mathbf{D}_\mu$ is the $N \times N$ diagonal matrix with the stationary distribution $\mathbf{d}_\mu$ on its diagonal. Thus, the key matrix that must be positive definite is $\mathbf{D}_\mu(\mathbf{I} - \gamma\mathbf{P}_\pi)$ and, unlike in the on-policy case, the distribution and the transition probabilities do not match. We do not have an analog of (5), $\mathbf{P}_\pi^\top \mathbf{d}_\mu \neq \mathbf{d}_\mu$, and in fact the column sums may be negative and the matrix not positive definite, in which case divergence of the parameter is likely.

A simple $\theta \to 2\theta$ example of divergence that fits the setting in this section is shown in Figure 1. From each state there are two actions, left and right, which take the process to the left or right states. All the rewards are zero. As before, there is a single parameter $\theta$ and the single feature is 1 and 2 in the two states such that the approximate values are $\theta$ and $2\theta$ as shown. The behavior policy is to go left and right with equal probability from both states, such that equal time is spent on average in both states, $\mathbf{d}_\mu = (0.5, 0.5)^\top$. The target policy is to go right in both states. We seek to learn the value from each state given that the right action is continually taken. The transition probability matrix for this example is:

$$\mathbf{P}_\pi = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}.$$

The key matrix is

$$\mathbf{D}_\mu(\mathbf{I} - \gamma\mathbf{P}_\pi) = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \times \begin{bmatrix} 1 & -0.9 \\ 0 & 0.1 \end{bmatrix} = \begin{bmatrix} 0.5 & -0.45 \\ 0 & 0.05 \end{bmatrix}. \tag{9}$$

We can see an immediate indication that the key matrix may not be positive definite in that the second column sums to a negative number. More definitively, one can show that it is not positive definite by multiplying it on both sides by $\mathbf{y} = \mathbf{\Phi} = (1, 2)^\top$:

$$\mathbf{\Phi}^\top \mathbf{D}_\mu(\mathbf{I} - \gamma\mathbf{P}_\pi)\mathbf{\Phi} = \begin{bmatrix} 1 & 2 \end{bmatrix} \times \begin{bmatrix} 0.5 & -0.45 \\ 0 & 0.05 \end{bmatrix} \times \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 & 2 \end{bmatrix} \times \begin{bmatrix} -0.4 \\ 0.1 \end{bmatrix} = -0.2.$$

That this is negative means that the key matrix is not positive definite. We have also calculated here the $\mathbf{A}$ matrix; it is this negative scalar, $\mathbf{A} = -0.2$. Clearly, this expected update and algorithm are not stable.

It is also easy to see the instability of this example more directly, without matrices. We know that only transitions under the right action cause updates, as $\rho_t$ will be zero for the others. Assume for concreteness that initially $\theta_t = 10$ and that $\alpha = 0.1$. On a right transition from the first state the update will be

$$\begin{aligned} \theta_{t+1} &= \theta_t + \rho_t \alpha \left( R_{t+1} + \gamma \boldsymbol{\theta}_t^\top \boldsymbol{\phi}_{t+1} - \boldsymbol{\theta}_t^\top \boldsymbol{\phi}_t \right) \boldsymbol{\phi}_t \\ &= 10 + 2 \cdot 0.1 \left( 0 + 0.9 \cdot 10 \cdot 2 - 10 \cdot 1 \right) 1 \\ &= 10 + 1.6, \end{aligned}$$

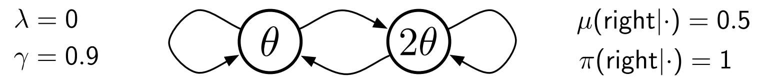

$\lambda = 0$
$\gamma = 0.9$
$\theta$
$2\theta$
$\mu(\text{right}|\cdot) = 0.5$
$\pi(\text{right}|\cdot) = 1$

Figure 1: $\theta \to 2\theta$ example without a terminal state.

whereas, on a $\mathsf{right}$ transition from the second state the update will be

$$\begin{aligned}\theta_{t+1} &= \theta_t + \rho_t\alpha\left(R_{t+1} + \gamma\boldsymbol{\theta}_t^\top\boldsymbol{\phi}_{t+1} - \boldsymbol{\theta}_t^\top\boldsymbol{\phi}_t\right)\boldsymbol{\phi}_t \\ &= 10 + 2\cdot 0.1\,(0 + 0.9\cdot 10\cdot 2 - 10\cdot 2)\,2 \\ &= 10 - 0.8.\end{aligned}$$

These two transitions occur equally often, so the net change will be positive. That is, $\theta$ will increase, moving farther from its correct value, zero. Everything is linear in $\theta$, so the next time around, with a larger starting $\theta$, the increase in $\theta$ will be larger still, and divergence occurs. A smaller value of $\alpha$ would not prevent divergence, only reduce its rate.

## 4. Off-policy Stability of Emphatic TD(0)

The deep reason for the difficulty of off-policy learning is that the behavior policy may take the process to a distribution of states different from that which would be encountered under the target policy, yet the states might appear to be the same or similar because of function approximation. Earlier work by Precup, Sutton and Dasgupta (2001) attempted to completely correct for the different state distribution using importance sampling ratios to reweight the states encountered. It is theoretically possible to convert the state weighting from $d_\mu$ to $d_\pi$ using the product of all importance sampling ratios from time 0, but in practice this approach has extremely high variance and is infeasible for the continuing (non-episodic) case. It works in theory because after converting the weighting the key matrix is $\mathbf{D}_\pi(\mathbf{I} - \gamma\mathbf{P}_\pi)$ again, which we know to be positive definite.

Most subsequent works abandoned the idea of completely correcting for the state distribution. For example, the work on gradient-TD methods (e.g., Sutton et al. 2009, Maei 2011) seeks to minimize the mean-squared projected Bellman error weighted by $d_\mu$. We call this an *excursion* setting because we can think of the contemplated switch to the target policy as an excursion from the steady-state distribution of the behavior policy, $d_\mu$. The excursions would start from $d_\mu$ and then follow $\pi$ until termination, followed by a resumption of $\mu$ and thus a gradual return to $d_\mu$. Of course these excursions never actually occur during off-policy learning, they are just contemplated, and thus the state distribution in fact never leaves $d_\mu$. It is the excursion view that we take in this paper, but still we use techniques similar to those introduced by Precup et al. (2001) to determine an emphasis weighting that corrects for the state distribution, only toward a different goal (see also Kolter 2011).

The excursion notion suggests a different weighting of TD(0) updates. We consider that at every time step we are beginning a new contemplated excursion from the current state. The excursion thus would begin in a state sampled from $d_\mu$. If an excursion started it would pass through a sequence of subsequent states and actions prior to termination. Some of the actions that are actually taken (under $\mu$) are relatively likely to occur under the target policy as compared to the behavior policy, while others are relatively unlikely; the corresponding states can be appropriately reweighted based on importance sampling ratios. Thus, there will still be a product of importance sampling ratios, but only since the beginning of the excursion, and the variance will also be tamped down by the discounting; the variance will be much less than in the earlier approach. In the simplest case of an off-policy emphatic algorithm, the update at time $t$ is emphasized or de-emphasized proportional to a new scalar variable $F_t$, defined by $F_0 = 1$ and

$$F_t \doteq \gamma \rho_{t-1} F_{t-1} + 1, \qquad \forall t > 0, \tag{10}$$

which we call the *followon trace*. Specifically, we define *emphatic TD(0)* by the following update:

$$\begin{aligned}\boldsymbol{\theta}_{t+1} &\doteq \boldsymbol{\theta}_t + \alpha F_t \rho_t \left( R_{t+1} + \gamma \boldsymbol{\theta}_t^\top \boldsymbol{\phi}_{t+1} - \boldsymbol{\theta}_t^\top \boldsymbol{\phi}_t \right) \boldsymbol{\phi}_t \qquad (11)\\ &= \boldsymbol{\theta}_t + \alpha \Big( \underbrace{F_t \rho_t R_{t+1} \boldsymbol{\phi}_t}_{\mathbf{b}_t} - \underbrace{F_t \rho_t \boldsymbol{\phi}_t \left( \boldsymbol{\phi}_t - \gamma \boldsymbol{\phi}_{t+1} \right)^\top}_{\mathbf{A}_t} \boldsymbol{\theta}_t \Big).\end{aligned}$$

Emphatic TD(0)'s $\mathbf{A}$ matrix is

$$\begin{aligned}\mathbf{A} = \lim_{t\to\infty} \mathbb{E}[\mathbf{A}_t] &= \lim_{t\to\infty} \mathbb{E}_\mu \Big[ F_t \rho_t \boldsymbol{\phi}_t \left( \boldsymbol{\phi}_t - \gamma \boldsymbol{\phi}_{t+1} \right)^\top \Big] \\ &= \sum_s d_\mu(s) \lim_{t\to\infty} \mathbb{E}_\mu \Big[ F_t \rho_t \boldsymbol{\phi}_t \left( \boldsymbol{\phi}_t - \gamma \boldsymbol{\phi}_{t+1} \right)^\top \Big| S_t = s \Big] \\ &= \sum_s d_\mu(s) \lim_{t\to\infty} \mathbb{E}_\mu [F_t | S_t = s] \, \mathbb{E}_\mu \Big[ \rho_t \boldsymbol{\phi}_t \left( \boldsymbol{\phi}_t - \gamma \boldsymbol{\phi}_{t+1} \right)^\top \Big| S_t = s \Big]\end{aligned}$$

(because, given $S_t$, $F_t$ is independent of $\rho_t \boldsymbol{\phi}_t \left( \boldsymbol{\phi}_t - \gamma \boldsymbol{\phi}_{t+1} \right)^\top$)

$$\begin{aligned}&= \sum_s \underbrace{d_\mu(s) \lim_{t\to\infty} \mathbb{E}_\mu [F_t | S_t = s]}_{f(s)} \, \mathbb{E}_\mu \Big[ \rho_k \boldsymbol{\phi}_k \left( \boldsymbol{\phi}_k - \gamma \boldsymbol{\phi}_{k+1} \right)^\top \Big| S_k = s \Big] \\ &= \sum_s f(s) \mathbb{E}_\pi \Big[ \boldsymbol{\phi}_k \left( \boldsymbol{\phi}_k - \gamma \boldsymbol{\phi}_{k+1} \right)^\top \Big| S_k = s \Big] \qquad \text{(by (7))} \\ &= \sum_s f(s) \, \boldsymbol{\phi}(s) \left( \boldsymbol{\phi}(s) - \gamma \sum_{s'} [\mathbf{P}_\pi]_{ss'} \boldsymbol{\phi}(s') \right)^\top \\ &= \boldsymbol{\Phi}^\top \mathbf{F} (\mathbf{I} - \gamma \mathbf{P}_\pi) \boldsymbol{\Phi},\end{aligned}$$

where $\mathbf{F}$ is a diagonal matrix with diagonal elements $f(s) \doteq d_\mu(s) \lim_{t\to\infty} \mathbb{E}_\mu[F_t | S_t = s]$, which we assume exists. As we show later, the vector $\mathbf{f} \in \mathbb{R}^N$ with components $[\mathbf{f}]_s \doteq f(s)$ can be written as

$$\begin{aligned}\mathbf{f} &= \mathbf{d}_\mu + \gamma \mathbf{P}_\pi^\top \mathbf{d}_\mu + \left( \gamma \mathbf{P}_\pi^\top \right)^2 \mathbf{d}_\mu + \cdots \qquad (12)\\ &= \left( \mathbf{I} - \gamma \mathbf{P}_\pi^\top \right)^{-1} \mathbf{d}_\mu. \qquad (13)\end{aligned}$$

The key matrix is $\mathbf{F} \left( \mathbf{I} - \gamma \mathbf{P}_\pi \right)$, and the vector of its column sums is

$$\begin{aligned}\mathbf{1}^\top \mathbf{F} (\mathbf{I} - \gamma \mathbf{P}_\pi) &= \mathbf{f}^\top (\mathbf{I} - \gamma \mathbf{P}_\pi) \\ &= \mathbf{d}_\mu^\top (\mathbf{I} - \gamma \mathbf{P}_\pi)^{-1} (\mathbf{I} - \gamma \mathbf{P}_\pi) \qquad \text{(using (13))} \\ &= \mathbf{d}_\mu^\top,\end{aligned}$$

all components of which are positive. Thus, the key matrix and the $\mathbf{A}$ matrix are positive definite and the algorithm is stable. Emphatic TD(0) is the simplest TD algorithm with linear function approximation proven to be stable under off-policy training.

The $\theta \rightarrow 2\theta$ example presented earlier (Figure 1) provides some insight into how replacing $\mathbf{D}_\mu$ by $\mathbf{F}$ changes the key matrix to make it positive definite. In general, $\mathbf{f}$ is the expected number of time steps that would be spent in each state during an excursion starting from the behavior distribution $\mathbf{d}_\mu$. From (12), it is $\mathbf{d}_\mu$ plus where you would get to in one step from $\mathbf{d}_\mu$, plus where you would get to in two steps, etc., with appropriate discounting. In the example, excursions under the target policy take you to the second state ($2\theta$) and leave you there. Thus you are only in the first state ($\theta$) if you start there, and only for one step, so $f(1) = d_\mu(1) = 0.5$. For the second state, you can either start there, with probability 0.5, or you can get there on the second step (certain except for discounting), with probability 0.9, or on the third step, with probability $0.9^2$, etc, so $f(2) = 0.5 + 0.9 + 0.9^2 + 0.9^3 + \cdots = 0.5 + 0.9 \cdot 10 = 9.5$. Thus, the key matrix is now

$$\mathbf{F}(\mathbf{I} - \gamma \mathbf{P}_\pi) = \begin{bmatrix} 0.5 & 0 \\ 0 & 9.5 \end{bmatrix} \times \begin{bmatrix} 1 & -0.9 \\ 0 & 0.1 \end{bmatrix} = \begin{bmatrix} 0.5 & -0.45 \\ 0 & 0.95 \end{bmatrix}.$$

Note that because $\mathbf{F}$ is a diagonal matrix, its only effect is to scale the rows. Here it emphasizes the lower row by more than a factor of 10 compared to the upper row, thereby causing the key matrix to have positive column sums and be positive definite (cf. (9)). The $\mathbf{F}$ matrix emphasizes the second state, which would occur much more often under the target policy than it does under the behavior policy.

## 5. The General Case

We turn now to a very general case of off-policy learning with linear function approximation. The objective is still to evaluate a policy $\pi$ from a single trajectory under a different policy $\mu$, but now the value of a state is defined not with respect to a constant discount rate $\gamma \in [0, 1]$, but with respect to a discount rate that varies from state to state according to a *discount function* $\gamma : \mathcal{S} \rightarrow [0, 1]$ such that $\prod_{k=1}^{\infty} \gamma(S_{t+k}) = 0, \text{w.p.1}, \forall t$. That is, our approximation is still defined by (2), but now (3) is replaced by

$$G_t \doteq R_{t+1} + \gamma(S_{t+1}) R_{t+2} + \gamma(S_{t+1}) \gamma(S_{t+2}) R_{t+3} + \cdots . \tag{14}$$

State-dependent discounting specifies a temporal envelope within which received rewards are accumulated. If $\gamma(S_k) = 0$, then the time of accumulation is fully terminated at step $k > t$, and if $\gamma(S_k) < 1$, then it is partially terminated. We call both of these *soft termination* because they are like the termination of an episode, but the actual trajectory is not affected. Soft termination ends the accumulation of rewards into a return, but the state transitions continue oblivious to the termination. Soft termination with state-dependent termination is essential for learning models of options (Sutton et al. 1999) and other applications.

Soft termination is particularly natural in the excursion setting, where it makes it easy to define excursions of finite and definite duration. For example, consider the deterministic MDP shown in Figure 2. There are five states, three of which do not discount at all, $\gamma(s) = 1$, and are shown as circles, and two of which cause complete soft termination, $\gamma(s) = 0$, and are shown as squares. The terminating states do not end anything other

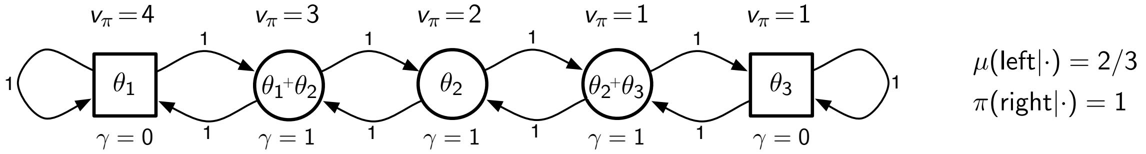


Figure 2: A 5-state chain MDP with soft-termination states at each end.

than the return; actions are still selected in them and, dependent on the action selected, they transition to next states indefinitely without end. In this MDP there are two actions, left and right, which deterministically cause transitions to the left or right except at the edges, where there may be a self transition. The reward on all transitions is +1. The behavior policy is to select left 2/3rds of the time in all states, which causes more time to be spent in states on the left than on the right. The stationary distribution can be shown to be $\mathbf{d}_\mu \approx (0.52, 0.26, 0.13, 0.06, 0.03)^\top$; more than half of the time steps are spent in the leftmost terminating state.

Consider the target policy $\pi$ that selects the right action from all states. The correct value $v_\pi(s)$ of each state $s$ is written above it in the figure. For both of the two rightmost states, the right action results in a reward of 1 and an immediate termination, so their values are both 1. For the middle state, following $\pi$ (selecting right repeatedly) yields two rewards of 1 prior to termination. There is no discounting ($\gamma=1$) prior to termination, so the middle state's value is 2, and similarly the values go up by 1 for each state to its left, as shown. These are the correct values. The approximate values depend on the parameter vector $\boldsymbol{\theta}_t$ as suggested by the expressions shown inside each state in the figure. These expressions use the notation $\theta_i$ to denote the $i$th *component* of the current parameter vector $\boldsymbol{\theta}_t$. In this example, there are five states and only three parameters, so it is unlikely, and indeed impossible, to represent $v_\pi$ exactly. We will return to this example later in the paper.

In addition to enabling definitive termination, as in this example, state-dependent discounting enables a much wider range of predictive questions to be expressed in the form of a value function (Sutton et al. 2011, Modayil, White & Sutton 2014, Sutton, Rafols & Koop 2006), including option models (Sutton, Precup & Singh 1999, Sutton 1995). For example, with state-dependent discounting one can formulate questions both about what will happen during a way of behaving and what will be true at its end. A general representation for predictions is a key step toward the goal of representing world knowledge in verifiable predictive terms (Sutton 2009, 2012). The general form is also useful just because it enables us to treat uniformly many of the most important episodic and continuing special cases of interest.

A second generalization, developed for the first time in this paper, is to explicitly specify the states at which we are most interested is obtaining accurate estimates of value. Recall that in parametric function approximation there are typically many more states than parameters ($N \gg n$), and thus it is usually not possible for the value estimates at all states to be exactly correct. Valuing some states more accurately usually means valuing others less accurately, at least asymptotically. In the tabular case where much of the theory of reinforcement learning originated, this tradeoff is not an issue because the estimates of each state are independent of each other, but with function approximation it is necessary to specify relative interest in order to make the problem well defined. Nevertheless, in the function approximation case little attention has been paid in the literature to specifing the relative importance of different states (an exception is Thomas 2014), though there are intimations of this in the initiation set of options (Sutton et al. 1999). In the past it was typically assumed that we were interested in valuing states in direct proportion to how often they occur, but this is not always the case. For example, in episodic problems we often care primarily about the value of the first state, or of earlier states generally (Thomas 2014). Here we allow the user to specify the relative interest in each state with a nonnegative *interest function* $i : \mathcal{S} \to [0, \infty)$. Formally, our objective is to minimize the Mean Square Value Error (MSVE) with states weighted both by how often they occur and by our interest in them:

$$\text{MSVE}(\boldsymbol{\theta}) \doteq \sum_{s \in \mathcal{S}} d_\mu(s) i(s) \Big( v_\pi(s) - \boldsymbol{\theta}^\top \boldsymbol{\phi}(s) \Big)^2. \tag{15}$$

For example, in the 5-state example in Figure 2, we could choose $i(s) = 1, \forall s \in \mathcal{S}$, in which case we would be primarily interested in attaining low error in the states on the left side, which are visited much more often under the behavior policy. If we want to counter this, we might chose $i(s)$ larger for states toward the right. Of course, with parametric function approximation we presumably do not have access to the states as individuals, but certainly we could set $i(s)$ as a function of the features in $s$. In this example, choosing $i(s) = 1 + \phi_2(s) + 2\phi_3(s)$ (where $\phi_i(s)$ denotes the $i$th component of $\boldsymbol{\phi}(s)$) would shift the focus on accuracy to the states on the right, making it substantially more balanced.

The third and final generalization that we introduce in this section is general bootstrapping. Conventional TD($\lambda$) uses a bootstrapping parameter $\lambda \in [0, 1]$; we generalize this to a *bootstrapping function* $\lambda : \mathcal{S} \to [0, 1]$ specifying a potentially different degree of bootstrapping, $1 - \lambda(s)$, for each state $s$. General bootstrapping of this form has been partially developed in several previous works (Sutton 1995, Sutton & Barto 1998, Maei & Sutton 2010, Sutton et al. 2014, cf. Yu 2012). As a notational shorthand, let us use $\lambda_t \doteq \lambda(S_t)$ and $\gamma_t \doteq \gamma(S_t)$. Then we can define a general notion of bootstrapped return, the $\lambda$-return with state-dependent bootstrapping and discounting:

$$G_t^\lambda \doteq R_{t+1} + \gamma_{t+1} \Big[ (1 - \lambda_{t+1}) \boldsymbol{\theta}_t^\top \boldsymbol{\phi}_{t+1} + \lambda_{t+1} G_{t+1}^\lambda \Big]. \tag{16}$$

The $\lambda$-return plays a key role in the theoretical understanding of TD methods, in particular, in their forward views (Sutton & Barto 1998, Sutton, Mahmood, Precup & van Hasselt 2014). In the forward view, $G_t^\lambda$ is thought of as the target for the update at time $t$, even though it is not available until many steps later (when complete termination $\gamma(S_k) = 0$ has occurred for the first time for some $k > t$).

Given these generalizations, we can now specify our final new algorithm, *emphatic TD($\lambda$)*, by the following four equations, for $t \geq 0$:

$$\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha \Big( R_{t+1} + \gamma_{t+1} \boldsymbol{\theta}_t^\top \boldsymbol{\phi}_{t+1} - \boldsymbol{\theta}_t^\top \boldsymbol{\phi}_t \Big) \mathbf{e}_t \tag{17}$$

$$\mathbf{e}_t \doteq \rho_t \left( \gamma_t \lambda_t \mathbf{e}_{t-1} + M_t \boldsymbol{\phi}_t \right), \quad \text{with } \mathbf{e}_{-1} \doteq \mathbf{0} \tag{18}$$

$$M_t \doteq \lambda_t \, i(S_t) + (1 - \lambda_t) F_t \tag{19}$$

$$F_t \doteq \rho_{t-1} \gamma_t F_{t-1} + i(S_t), \quad \text{with } F_0 \doteq i(S_0), \tag{20}$$

where $F_t \geq 0$ is a scalar memory called the *followon trace*. The quantity $M_t \geq 0$ is termed the *emphasis* on step $t$. Note that, if desired, $M_t$ can be removed from the algorithm by substituting its definition (19) into (18).

## 6. Off-policy Stability of Emphatic TD($\lambda$)

As usual, to analyze the stability of the new algorithm we examine its $\mathbf{A}$ matrix. The stochastic update can be written:

$$\begin{aligned}\boldsymbol{\theta}_{t+1} &\doteq \boldsymbol{\theta}_t + \alpha\left(R_{t+1} + \gamma_{t+1}\boldsymbol{\theta}_t^\top \boldsymbol{\phi}_{t+1} - \boldsymbol{\theta}_t^\top \boldsymbol{\phi}_t\right)\mathbf{e}_t \\ &= \boldsymbol{\theta}_t + \alpha\Big(\underbrace{\mathbf{e}_t R_{t+1}}_{\mathbf{b}_t} - \underbrace{\mathbf{e}_t\left(\boldsymbol{\phi}_t - \gamma_{t+1}\boldsymbol{\phi}_{t+1}\right)^\top}_{\mathbf{A}_t}\boldsymbol{\theta}_t\Big).\end{aligned}$$

Thus,

$$\begin{aligned}\mathbf{A} = \lim_{t\to\infty}\mathbb{E}[\mathbf{A}_t] &= \lim_{t\to\infty}\mathbb{E}_\mu\Big[\mathbf{e}_t\left(\boldsymbol{\phi}_t - \gamma_{t+1}\boldsymbol{\phi}_{t+1}\right)^\top\Big] \\ &= \sum_s d_\mu(s)\lim_{t\to\infty}\mathbb{E}_\mu\Big[\mathbf{e}_t\left(\boldsymbol{\phi}_t - \gamma_{t+1}\boldsymbol{\phi}_{t+1}\right)^\top\Big| S_t = s\Big] \\ &= \sum_s d_\mu(s)\lim_{t\to\infty}\mathbb{E}_\mu\Big[\rho_t\left(\gamma_t\lambda_t\mathbf{e}_{t-1} + M_t\boldsymbol{\phi}_t\right)\left(\boldsymbol{\phi}_t - \gamma_{t+1}\boldsymbol{\phi}_{t+1}\right)^\top\Big| S_t = s\Big] \\ &= \sum_s d_\mu(s)\lim_{t\to\infty}\mathbb{E}_\mu[(\gamma_t\lambda_t\mathbf{e}_{t-1} + M_t\boldsymbol{\phi}_t)|S_t = s]\,\mathbb{E}_\mu\Big[\rho_t(\boldsymbol{\phi}_t - \gamma_{t+1}\boldsymbol{\phi}_{t+1})^\top\Big| S_t = s\Big] \\ &\qquad \text{(because, given } S_t,\ \mathbf{e}_{t-1} \text{ and } M_t \text{ are independent of } \rho_t(\boldsymbol{\phi}_t - \gamma_{t+1}\boldsymbol{\phi}_{t+1})^\top) \\ &= \sum_s \underbrace{d_\mu(s)\lim_{t\to\infty}\mathbb{E}_\mu[(\gamma_t\lambda_t\mathbf{e}_{t-1} + M_t\boldsymbol{\phi}_t)|S_t = s]}_{\mathbf{e}(s)\in\mathbb{R}^n}\,\mathbb{E}_\mu\Big[\rho_k(\boldsymbol{\phi}_k - \gamma_{k+1}\boldsymbol{\phi}_{k+1})^\top\Big| S_k = s\Big] \\ &= \sum_s \mathbf{e}(s)\mathbb{E}_\pi[\boldsymbol{\phi}_k - \gamma_{k+1}\boldsymbol{\phi}_{k+1}|S_k = s]^\top \qquad \text{(by (7))} \\ &= \sum_s \mathbf{e}(s)\left(\boldsymbol{\phi}(s) - \sum_{s'}[\mathbf{P}_\pi]_{ss'}\gamma(s')\boldsymbol{\phi}(s')\right)^\top \\ &= \mathbf{E}(\mathbf{I} - \mathbf{P}_\pi\boldsymbol{\Gamma})\boldsymbol{\Phi}, \qquad (21)\end{aligned}$$

where $\mathbf{E}$ is an $N \times n$ matrix $\mathbf{E}^\top \doteq [\mathbf{e}(1), \cdots, \mathbf{e}(N)]$, and $\mathbf{e}(s) \in \mathbb{R}^n$ is defined by[5]:

$$\begin{aligned}\mathbf{e}(s) &\doteq d_\mu(s)\lim_{t\to\infty}\mathbb{E}_\mu[\gamma_t\lambda_t\mathbf{e}_{t-1} + M_t\boldsymbol{\phi}_t|S_t = s] \qquad \text{(assuming this exists)} \\ &= \underbrace{d_\mu(s)\lim_{t\to\infty}\mathbb{E}_\mu[M_t|S_t = s]}_{m(s)}\,\boldsymbol{\phi}(s) + \gamma(s)\lambda(s)d_\mu(s)\lim_{t\to\infty}\mathbb{E}_\mu[\mathbf{e}_{t-1}|S_t = s] \\ &= m(s)\boldsymbol{\phi}(s) + \gamma(s)\lambda(s)d_\mu(s)\lim_{t\to\infty}\sum_{\bar{s},\bar{a}}\mathbb{P}\{S_{t-1} = \bar{s}, A_{t-1} = \bar{a}|S_t = s\}\,\mathbb{E}_\mu[\mathbf{e}_{t-1}|S_{t-1} = \bar{s}, A_{t-1} = \bar{a}]\end{aligned}$$

[5] Note that this is a slight abuse of notation; $\mathbf{e}_t$ is a vector random variable, one per time step, and $\mathbf{e}(s)$ is a vector expectation, one per state.

$$= m(s)\boldsymbol{\phi}(s) + \gamma(s)\lambda(s)d_\mu(s)\sum_{\bar{s},\bar{a}} \frac{d_\mu(\bar{s})\mu(\bar{a}|\bar{s})p(s|\bar{s},\bar{a})}{d_\mu(s)} \lim_{t\to\infty} \mathbb{E}_\mu[\mathbf{e}_{t-1}|S_{t-1}=\bar{s}, A_{t-1}=\bar{a}]$$

(using the definition of a conditional probability, a.k.a. Bayes rule)

$$= m(s)\boldsymbol{\phi}(s)+\gamma(s)\lambda(s)\sum_{\bar{s},\bar{a}} d_\mu(\bar{s})\mu(\bar{a}|\bar{s})p(s|\bar{s},\bar{a})\frac{\pi(\bar{a}|\bar{s})}{\mu(\bar{a}|\bar{s})}\lim_{t\to\infty}\mathbb{E}_\mu[\gamma_{t-1}\lambda_{t-1}\mathbf{e}_{t-2}+M_{t-1}\boldsymbol{\phi}_{t-1}|S_{t-1}=\bar{s}]$$

$$= m(s)\boldsymbol{\phi}(s) + \gamma(s)\lambda(s)\sum_{\bar{s}}\left(\sum_{\bar{a}}\pi(\bar{a}|\bar{s})p(s|\bar{s},\bar{a})\right)\mathbf{e}(\bar{s})$$

$$= m(s)\boldsymbol{\phi}(s) + \gamma(s)\lambda(s)\sum_{\bar{s}}[\mathbf{P}_\pi]_{\bar{s}s}\mathbf{e}(\bar{s}).$$

We now introduce three $N \times N$ diagonal matrices: $\mathbf{M}$, which has the $m(s) \doteq d_\mu(s)\lim_{t\to\infty} \mathbb{E}_\mu[M_t|S_t=s]$ on its diagonal; $\mathbf{\Gamma}$, which has the $\gamma(s)$ on its diagonal; and $\mathbf{\Lambda}$, which has the $\lambda(s)$ on its diagonal. With these we can write the equation above entirely in matrix form, as

$$\begin{aligned}\mathbf{E}^\top &= \mathbf{\Phi}^\top\mathbf{M} + \mathbf{E}^\top\mathbf{P}_\pi\mathbf{\Gamma\Lambda}\\ &= \mathbf{\Phi}^\top\mathbf{M} + \mathbf{\Phi}^\top\mathbf{M}\mathbf{P}_\pi\mathbf{\Gamma\Lambda} + \mathbf{\Phi}^\top\mathbf{M}(\mathbf{P}_\pi\mathbf{\Gamma\Lambda})^2 + \cdots\\ &= \mathbf{\Phi}^\top\mathbf{M}(\mathbf{I}-\mathbf{P}_\pi\mathbf{\Gamma\Lambda})^{-1}.\end{aligned}$$

Finally, combining this equation with (21) we obtain

$$\mathbf{A} = \mathbf{\Phi}^\top\mathbf{M}(\mathbf{I}-\mathbf{P}_\pi\mathbf{\Gamma\Lambda})^{-1}(\mathbf{I}-\mathbf{P}_\pi\mathbf{\Gamma})\mathbf{\Phi}, \tag{22}$$

and through similar steps one can also obtain emphatic TD($\lambda$)'s $\mathbf{b}$ vector,

$$\mathbf{b} = \mathbf{E}\mathbf{r}_\pi = \mathbf{\Phi}^\top\mathbf{M}(\mathbf{I}-\mathbf{P}_\pi\mathbf{\Gamma\Lambda})^{-1}\mathbf{r}_\pi, \tag{23}$$

where $\mathbf{r}_\pi$ is the $N$-vector of expected immediate rewards from each state under $\pi$.

Emphatic TD($\lambda$)'s key matrix, then, is $\mathbf{M}(\mathbf{I}-\mathbf{P}_\pi\mathbf{\Gamma\Lambda})^{-1}(\mathbf{I}-\mathbf{P}_\pi\mathbf{\Gamma})$. To prove that it is positive definite we will follow the same strategy as we did for emphatic TD(0). The first step will be to write the last part of the key matrix in the form of the identity matrix minus a probability matrix. To see how this can be done, consider a slightly different setting in which actions are taken according to $\pi$, and in which $1-\gamma(s)$ and $1-\lambda(s)$ are considered probabilities of ending by terminating and by bootstrapping, respectively. That is, for any starting state, a trajectory involves a state transition according to $\mathbf{P}_\pi$, possibly terminating according to $\mathbf{I}-\mathbf{\Gamma}$, then possibly ending with a bootstrapping event according to $\mathbf{I}-\mathbf{\Lambda}$, and then, if neither of these occur, continuing with another state transition and more chances to end, and so on until an ending of one of the two kinds occurs. For any start state $i \in \mathcal{S}$, consider the probability that the trajectory ends in state $j \in \mathcal{S}$ with an ending event of the bootstrapping kind (according to $\mathbf{I}-\mathbf{\Lambda}$). Let $\mathbf{P}_\pi^\lambda$ be the matrix with this probability as its

$ij$th component. This matrix can be written

$$\begin{aligned}\mathbf{P}_\pi^\lambda &= \mathbf{P}_\pi\boldsymbol{\Gamma}(\mathbf{I}-\boldsymbol{\Lambda}) + \mathbf{P}_\pi\boldsymbol{\Gamma}\boldsymbol{\Lambda}\mathbf{P}_\pi\boldsymbol{\Gamma}(\mathbf{I}-\boldsymbol{\Lambda}) + \mathbf{P}_\pi\boldsymbol{\Gamma}(\boldsymbol{\Lambda}\mathbf{P}_\pi\boldsymbol{\Gamma})^2(\mathbf{I}-\boldsymbol{\Lambda}) + \cdots \\ &= \left(\sum_{k=0}^{\infty}(\mathbf{P}_\pi\boldsymbol{\Gamma}\boldsymbol{\Lambda})^k\right)\mathbf{P}_\pi\boldsymbol{\Gamma}(\mathbf{I}-\boldsymbol{\Lambda}) \\ &= (\mathbf{I}-\mathbf{P}_\pi\boldsymbol{\Gamma}\boldsymbol{\Lambda})^{-1}\mathbf{P}_\pi\boldsymbol{\Gamma}(\mathbf{I}-\boldsymbol{\Lambda}). \\ &= (\mathbf{I}-\mathbf{P}_\pi\boldsymbol{\Gamma}\boldsymbol{\Lambda})^{-1}(\mathbf{P}_\pi\boldsymbol{\Gamma}-\mathbf{P}_\pi\boldsymbol{\Gamma}\boldsymbol{\Lambda}) \\ &= (\mathbf{I}-\mathbf{P}_\pi\boldsymbol{\Gamma}\boldsymbol{\Lambda})^{-1}(\mathbf{P}_\pi\boldsymbol{\Gamma}-\mathbf{I}+\mathbf{I}-\mathbf{P}_\pi\boldsymbol{\Gamma}\boldsymbol{\Lambda}) \\ &= \mathbf{I}-(\mathbf{I}-\mathbf{P}_\pi\boldsymbol{\Gamma}\boldsymbol{\Lambda})^{-1}(\mathbf{I}-\mathbf{P}_\pi\boldsymbol{\Gamma}),\end{aligned}$$

or

$$\mathbf{I}-\mathbf{P}_\pi^\lambda = (\mathbf{I}-\mathbf{P}_\pi\boldsymbol{\Gamma}\boldsymbol{\Lambda})^{-1}(\mathbf{I}-\mathbf{P}_\pi\boldsymbol{\Gamma}). \tag{24}$$

It follows then that $\mathbf{M}(\mathbf{I}-\mathbf{P}_\pi^\lambda) = \mathbf{M}(\mathbf{I}-\mathbf{P}_\pi\boldsymbol{\Gamma}\boldsymbol{\Lambda})^{-1}(\mathbf{I}-\mathbf{P}_\pi\boldsymbol{\Gamma})$ is another way of writing emphatic TD($\lambda$)'s key matrix (cf. (22)). This gets us considerably closer to our goal of proving that the key matrix is positive definite. It is now immediate that its diagonal entries are nonnegative and that its off diagonal entries are nonpositive. It is also immediate that its row sums are nonnegative.

There remains what is typically the hardest condition to satisfy: that the column sums of the key matrix are positive. To show this we have to analyze $\mathbf{M}$, and to do that we first analyze the $N$-vector $\mathbf{f}$ with components $f(s) \doteq d_\mu(s)\lim_{t\to\infty}\mathbb{E}_\mu[F_t|S_t\!=\!s]$ (we assume that this limit and expectation exist). Analyzing $\mathbf{f}$ will also pay the debt we incurred in Section 4 when we claimed without proof that $\mathbf{f}$ (in the special case treated in that section) was as given by (13). In the general case:

$$\begin{aligned}f(s) &= d_\mu(s)\lim_{t\to\infty}\mathbb{E}_\mu[F_t|S_t\!=\!s] \\ &= d_\mu(s)\lim_{t\to\infty}\mathbb{E}_\mu[i(S_t)+\rho_{t-1}\gamma_t F_{t-1}|S_t\!=\!s] \qquad \text{(by (20))} \\ &= d_\mu(s)i(s) + d_\mu(s)\gamma(s)\lim_{t\to\infty}\sum_{\bar{s},\bar{a}}\mathbb{P}\{S_{t-1}\!=\!\bar{s},A_{t-1}\!=\!\bar{a}|S_t\!=\!s\}\,\frac{\pi(\bar{a}|\bar{s})}{\mu(\bar{a}|\bar{s})}\mathbb{E}_\mu[F_{t-1}|S_{t-1}\!=\!\bar{s}] \\ &= d_\mu(s)i(s) + d_\mu(s)\gamma(s)\sum_{\bar{s},\bar{a}}\frac{d_\mu(\bar{s})\mu(\bar{a}|\bar{s})p(s|\bar{s},\bar{a})}{d_\mu(s)}\frac{\pi(\bar{a}|\bar{s})}{\mu(\bar{a}|\bar{s})}\lim_{t\to\infty}\mathbb{E}_\mu[F_{t-1}|S_{t-1}\!=\!\bar{s}]\end{aligned}$$

(using the definition of a conditional probability, a.k.a. Bayes rule)

$$\begin{aligned}&= d_\mu(s)i(s) + \gamma(s)\sum_{\bar{s},\bar{a}}\pi(\bar{a}|\bar{s})p(s|\bar{s},\bar{a})d_\mu(\bar{s})\lim_{t\to\infty}\mathbb{E}_\mu[F_{t-1}|S_{t-1}\!=\!\bar{s}] \\ &= d_\mu(s)i(s) + \gamma(s)\sum_{\bar{s}}[\mathbf{P}_\pi]_{\bar{s}s}f(\bar{s}).\end{aligned}$$

This equation can be written in matrix-vector form, letting $\mathbf{i}$ be the $N$-vector with components $[\mathbf{i}]_s \doteq d_\mu(s)i(s)$:

$$\begin{aligned}\mathbf{f} &= \mathbf{i} + \boldsymbol{\Gamma}\mathbf{P}_\pi^\top\mathbf{f} \\ &= \mathbf{i} + \boldsymbol{\Gamma}\mathbf{P}_\pi^\top\mathbf{i} + (\boldsymbol{\Gamma}\mathbf{P}_\pi^\top)^2\mathbf{i} + \cdots \\ &= \left(\mathbf{I}-\boldsymbol{\Gamma}\mathbf{P}_\pi^\top\right)^{-1}\mathbf{i}.\end{aligned} \tag{25}$$

This proves (13), because there $i(s) \doteq 1, \forall s$ (thus $\mathbf{i} = \mathbf{d}_\mu$), and $\gamma(s) \doteq \gamma, \forall s$.

We are now ready to analyze $\mathbf{M}$, the diagonal matrix with the $m(s)$ on its diagonal:

$$\begin{aligned} m(s) &= d_\mu(s) \lim_{t\to\infty} \mathbb{E}_\mu[M_t|S_t\!=\!s] \\ &= d_\mu(s) \lim_{t\to\infty} \mathbb{E}_\mu[\lambda_t\, i(S_t) + (1-\lambda_t)F_t|S_t\!=\!s] && \text{(by (19))} \\ &= d_\mu(s)\lambda(s)i(s) + (1-\lambda(s))\, f(s), \end{aligned}$$

or, in matrix-vector form, letting $\mathbf{m}$ be the $N$-vector with components $m(s)$,

$$\begin{aligned} \mathbf{m} &= \mathbf{\Lambda i} + (\mathbf{I} - \mathbf{\Lambda})\mathbf{f} \\ &= \mathbf{\Lambda i} + (\mathbf{I} - \mathbf{\Lambda}) \left(\mathbf{I} - \mathbf{\Gamma P}_\pi^\top\right)^{-1} \mathbf{i} && \text{(using (25))} \\ &= \left[\mathbf{\Lambda}(\mathbf{I} - \mathbf{\Gamma P}_\pi^\top) + (\mathbf{I} - \mathbf{\Lambda})\right] (\mathbf{I} - \mathbf{\Gamma P}_\pi^\top)^{-1}\mathbf{i} \\ &= \left(\mathbf{I} - \mathbf{\Lambda\Gamma P}_\pi^\top\right) \left(\mathbf{I} - \mathbf{\Gamma P}_\pi^\top\right)^{-1} \mathbf{i} && (26) \\ &= \left(\mathbf{I} - \mathbf{P}_\pi^{\lambda\top}\right)^{-1} \mathbf{i}. && \text{(using (24))} \end{aligned}$$

Now we are ready for the final step of the proof, showing that all the columns of the key matrix $\mathbf{M}(\mathbf{I} - \mathbf{P}_\pi^\lambda)$ sum to a positive number. Using the result above, the vector of column sums is

$$\begin{aligned} \mathbf{1}^\top \mathbf{M}(\mathbf{I} - \mathbf{P}_\pi^\lambda) &= \mathbf{m}^\top (\mathbf{I} - \mathbf{P}_\pi^\lambda) \\ &= \mathbf{i}^\top (\mathbf{I} - \mathbf{P}_\pi^\lambda)^{-1} (\mathbf{I} - \mathbf{P}_\pi^\lambda) \\ &= \mathbf{i}^\top. \end{aligned}$$

If we further assume that $i(s) > 0, \forall s \in \mathcal{S}$, then the column sums are all positive, the key matrix is positive definite, and emphatic TD($\lambda$) and its expected update are stable. This result can be summarized in the following theorem, the main result of this paper, which we have just proved:

**Theorem 1 (Stability of Emphatic TD($\lambda$))** *For any*

- *Markov decision process* $\{S_t, A_t, R_{t+1}\}_{t=0}^\infty$ *with finite state and actions sets* $\mathcal{S}$ *and* $\mathcal{A}$,
- *behavior policy* $\mu$ *with a stationary invariant distribution* $d_\mu(s) > 0, \forall s \in \mathcal{S}$,
- *target policy* $\pi$ *with coverage, i.e., s.t., if* $\pi(a|s) > 0$*, then* $\mu(a|s) > 0$,
- *discount function* $\gamma : \mathcal{S} \to [0, 1]$ *s.t.* $\prod_{k=1}^\infty \gamma(S_{t+k}) = 0$*, w.p.1,* $\forall t > 0$,
- *bootstrapping function* $\lambda : \mathcal{S} \to [0, 1]$,
- *interest function* $i : \mathcal{S} \to (0, \infty)$,
- *feature function* $\boldsymbol{\phi} : \mathcal{S} \to \mathbb{R}^n$ *s.t. the matrix* $\mathbf{\Phi} \in \mathbb{R}^{|\mathcal{S}|\times n}$ *with the* $\boldsymbol{\phi}(s)$ *as its rows has linearly independent columns,*

*the* $\mathbf{A}$ *matrix of linear emphatic TD($\lambda$) (as given by (17–20), and assuming the existence of* $\lim_{t\to\infty}\mathbb{E}[F_t|S_t\!=s]$ *and* $\lim_{t\to\infty}\mathbb{E}[\mathbf{e}_t|S_t\!=s]$, $\forall s\in\mathcal{S}$*),*

$$\mathbf{A}=\lim_{t\to\infty}\mathbb{E}_\mu[\mathbf{A}_t]=\lim_{t\to\infty}\mathbb{E}_\mu\Big[\mathbf{e}_t\left(\boldsymbol{\phi}_t-\gamma_{t+1}\boldsymbol{\phi}_{t+1}\right)^\top\Big]=\mathbf{\Phi}^\top\mathbf{M}(\mathbf{I}-\mathbf{P}_\pi^\lambda)\mathbf{\Phi}, \tag{27}$$

*is positive definite. Thus the algorithm and its expected update are stable.*

As mentioned at the outset, stability is necessary but not always sufficient to guarantee convergence of the parameter vector $\boldsymbol{\theta}_t$. Yu (2015a,b) has recently built on our stability result to show that in fact emphatic TD($\lambda$) converges with probability one under a variety of step-size conditions. Convergence as anticipated is to the unique fixed point $\bar{\boldsymbol{\theta}}$ of the deterministic algorithm (6), in other words, to

$$\mathbf{A}\bar{\boldsymbol{\theta}}=\mathbf{b}\quad\text{ or }\quad\bar{\boldsymbol{\theta}}=\mathbf{A}^{-1}\mathbf{b}. \tag{28}$$

This solution can be characterized as a minimum (in fact, a zero) of the Projected Bellman Error (PBE, Sutton et al. 2009) using the $\lambda$-dependent Bellman operator $T^{(\lambda)}:\mathbb{R}^N\to\mathbb{R}^N$ (Tsitiklis & Van Roy 1997) and the weighting of states according to their emphasis. For our general case, we need a version of the $T^{(\lambda)}$ operator extended to state-dependent discounting and bootstrapping. This operator looks ahead to future states to the extent that they are bootstrapped from, that is, according to $\mathbf{P}_\pi^\lambda$, taking into account the reward received along the way. The appropriate operator, in vector form, is

$$T^{(\lambda)}\mathbf{v}\doteq(\mathbf{I}-\mathbf{P}_\pi\mathbf{\Gamma}\mathbf{\Lambda})^{-1}\mathbf{r}_\pi+\mathbf{P}_\pi^\lambda\mathbf{v}. \tag{29}$$

This operator is a contraction with fixed point $\mathbf{v}=v_\pi$. Recall that our approximate value function is $\mathbf{\Phi}\boldsymbol{\theta}$, and thus the difference between $\mathbf{\Phi}\boldsymbol{\theta}$ and $T^{(\lambda)}(\mathbf{\Phi}\boldsymbol{\theta})$ is a Bellman-error vector. The projection of this with respect to the feature matrix and the emphasis weighting is the emphasis-weighted PBE:

$$\begin{aligned}
\text{PBE}(\boldsymbol{\theta})&\doteq\Pi\left(T^{(\lambda)}(\mathbf{\Phi}\boldsymbol{\theta})-\mathbf{\Phi}\boldsymbol{\theta}\right)\\
&\doteq\mathbf{\Phi}(\mathbf{\Phi}^\top\mathbf{M}\mathbf{\Phi})^{-1}\mathbf{\Phi}^\top\mathbf{M}\left(T^{(\lambda)}(\mathbf{\Phi}\boldsymbol{\theta})-\mathbf{\Phi}\boldsymbol{\theta}\right) && \text{(see Sutton et al. 2009)}\\
&=\mathbf{\Phi}(\mathbf{\Phi}^\top\mathbf{M}\mathbf{\Phi})^{-1}\mathbf{\Phi}^\top\mathbf{M}\left((\mathbf{I}-\mathbf{P}_\pi\mathbf{\Gamma}\mathbf{\Lambda})^{-1}\mathbf{r}_\pi+\mathbf{P}_\pi^\lambda\mathbf{\Phi}\boldsymbol{\theta}-\mathbf{\Phi}\boldsymbol{\theta}\right) && \text{(by (29))}\\
&=\mathbf{\Phi}(\mathbf{\Phi}^\top\mathbf{M}\mathbf{\Phi})^{-1}\left(\mathbf{b}+\mathbf{\Phi}^\top\mathbf{M}(\mathbf{P}_\pi^\lambda-\mathbf{I})\mathbf{\Phi}\boldsymbol{\theta}\right) && \text{(by (23))}\\
&=\mathbf{\Phi}(\mathbf{\Phi}^\top\mathbf{M}\mathbf{\Phi})^{-1}\left(\mathbf{b}-\mathbf{A}\boldsymbol{\theta}\right). && \text{(by (27))}
\end{aligned}$$

From (28), it is immediate that this is vector zero at the fixed point $\bar{\boldsymbol{\theta}}$, thus $\text{PBE}(\bar{\boldsymbol{\theta}})=\mathbf{0}$.

## 7. Derivation of the Emphasis Algorithm

Emphatic algorithms are based on the idea that if we are updating a state by a TD method, then we should also update each state that it bootstraps from, in direct proportion. For example, suppose we decide to update the estimate at time $t$ with unit emphasis, perhaps because $i(S_t)=1$, and then at time $t+1$ we have $\gamma(S_{t+1})=1$ and $\lambda(S_{t+1})=0$. Because of the latter, we are fully bootstrapping from the value estimate at $t+1$ and thus we should also make an update of it with emphasis equal to $t$'s emphasis. If instead $\lambda(S_{t+1}) = 0.5$, then the update of the estimate at $t+1$ would gain a half unit of emphasis, and the remaining half would still be available to allocate to the updates of the estimate at $t+2$ or later times depending on their $\lambda$s. And of course there may be some emphasis allocated directly updating the estimate at $t+1$ if $i(S_{t+1}) > 0$. Discounting and importance sampling also have effects. At each step $t$, if $\gamma(S_t) < 1$, then there is some degree of termination and to that extent there is no longer any chance of bootstrapping from later time steps. Another way bootstrapping may be cut off is if $\rho_t = 0$ (a complete deviation from the target policy). More generally, if $\rho \neq 1$, then the opportunity for bootstrapping is scaled up or down proportionally.

It may seem difficult to work out precisely how each time step's estimates bootstrap from which later states' estimates for all cases. Fortunately, it has already been done. Equation 6 of the paper by Sutton, Mahmood, Precup, and van Hasselt (2014) specifies this in their "forward view" of off-policy TD($\lambda$) with general state-dependent discounting and bootstrapping. From this equation (and their (5)) it is easy to determine the degree to which the update of the value estimate at time $k$ bootstraps from (multiplicatively depends on) the value estimates of each subsequent time $t$. It is

$$\rho_k \left( \prod_{i=k+1}^{t-1} \gamma_i \lambda_i \rho_i \right) \gamma_t (1 - \lambda_t).$$

It follows then that the total emphasis on time $t$, $M_t$, should be the sum of this quantity for all times $k < t$, each times the emphasis $M_k$ for those earlier times, plus any intrinsic interest $i(S_t)$ in time $t$:

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

which is (19), where

$$\begin{aligned}
F_t &\doteq i(S_t) + \gamma_t \sum_{k=0}^{t-1} \rho_k M_k \prod_{i=k+1}^{t-1} \gamma_i \lambda_i \rho_i \\
&= i(S_t) + \gamma_t \left( \rho_{t-1} M_{t-1} + \sum_{k=0}^{t-2} \rho_k M_k \prod_{i=k+1}^{t-1} \gamma_i \lambda_i \rho_i \right) \\
&= i(S_t) + \gamma_t \left( \rho_{t-1} M_{t-1} + \rho_{t-1} \lambda_{t-1} \gamma_{t-1} \sum_{k=0}^{t-2} \rho_k M_k \prod_{i=k+1}^{t-2} \gamma_i \lambda_i \rho_i \right) \\
&= i(S_t) + \gamma_t \rho_{t-1} \left( \underbrace{\lambda_{t-1} i(S_{t-1}) + (1 - \lambda_{t-1}) F_{t-1}}_{M_{t-1}} + \lambda_{t-1} \gamma_{t-1} \sum_{k=0}^{t-2} \rho_k M_k \prod_{i=k+1}^{t-2} \gamma_i \lambda_i \rho_i \right)
\end{aligned}$$

$$= i(S_t) + \gamma_t \rho_{t-1} \left( F_{t-1} + \lambda_{t-1} \Bigg( -F_{t-1} + \underbrace{i(S_{t-1}) + \gamma_{t-1} \sum_{k=0}^{t-2} \rho_k M_k \prod_{i=k+1}^{t-2} \gamma_i \lambda_i \rho_i}_{F_{t-1}} \Bigg) \right)$$

$$= i(S_t) + \gamma_t \rho_{t-1} F_{t-1},$$

which is (20), completing the derivation of the emphasis algorithm.

## 8. Empirical Examples

In this section we present empirical results with example problems that verify and elucidate the formal results already presented. A thorough empirical comparison of emphatic TD($\lambda$) with other methods is beyond the scope of the present article.

The main focus in this paper, as in much previous theory of TD algorithms with function approximation, has been on the stability of the expected update. If an algorithm is unstable, as Q-learning and off-policy TD($\lambda$) are on Baird's (1995) counterexample, then there is no chance of its behaving in a satisfactory manner. On the other hand, even if the update is stable it may be of very high variance. Off-policy algorithms involve products of potentially an infinite number of importance-sampling ratios, which can lead to fluxuations of infinite variance.

As an example of what can happen, let's look again at the $\theta \to 2\theta$ problem shown in Figure 1 (and shown again in the upper left of Figure 3). Consider what happens to $F_t$ in this problem if we have interest only in the first state, and the right action happens to be taken on every step (i.e., $i(S_0) = 1$ then $i(S_t) = 0, \forall t > 0$, and $A_t = \mathsf{right}, \forall t \geq 0$). In this case, from (20),

$$F_t = \rho_{t-1}\gamma_t F_{t-1} + i(S_t) = \prod_{j=0}^{t-1} \rho_j \gamma = (2 \cdot 0.9)^t,$$

which of course goes to infinity as $t \to \infty$. On the other hand, the probability of this specific infinite action sequence is zero, and in fact $F_t$ will rarely take on very high values. In particular, the expected value of $F_t$ remains finite at

$$\begin{aligned} \mathbb{E}_\mu[F_t] &= 0.5 \cdot 2 \cdot 0.9 \cdot \mathbb{E}_\mu[F_{t-1}] + 0.5 \cdot 0 \cdot 0.9 \cdot \mathbb{E}_\mu[F_{t-1}] \\ &= 0.9 \cdot \mathbb{E}_\mu[F_{t-1}] \\ &= 0.9^t, \end{aligned}$$

which tends to zero as $t \to \infty$. Nevertheless, this problem is indeed a difficult case, as the *variance* of $F_t$ *is* infinite:

$$\begin{aligned} \mathrm{Var}[F_t] &= \mathbb{E}\left[F_t^2\right] - (\mathbb{E}[F_t])^2 \\ &= 0.5^t(2^t 0.9^t)^2 - (0.9^t)^2 \\ &= (0.9^2 \cdot 2)^t - (0.9^2)^t \\ &= 1.62^t - 0.81^t, \end{aligned}$$

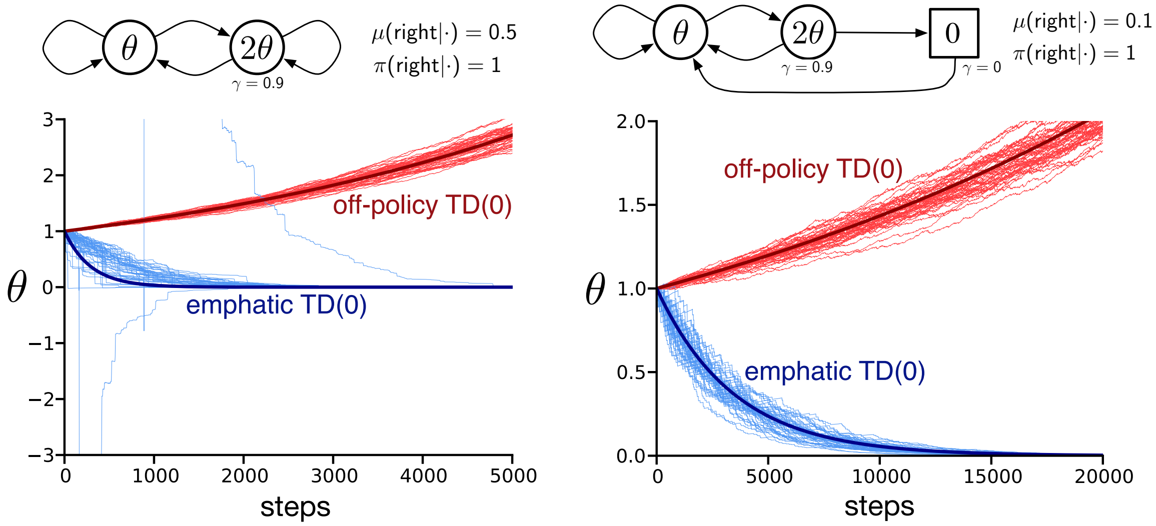


Figure 3: Emphatic TD approaches the correct value of zero, whereas conventional off-policy TD diverges, on fifty trajectories on the $\theta \rightarrow 2\theta$ problems shown above each graph. Also shown as a thick line is the trajectory of the deterministic expected-update algorithm (6). On the continuing problem (left) emphatic TD had occasional high variance deviations from zero.

which tends to $\infty$ as $t \rightarrow \infty$.

What does actually happen on this problem? The thin blue lines in Figure 3 (left) show the trajectories of the single parameter $\theta$ over time in 50 runs with this problem with $\lambda = 0$ and $\alpha = 0.001$, starting at $\theta = 1.0$. We see that most trajectories of emphatic TD(0) rapidly approached the correct value of $\theta = 0$, but a few made very large steps away from zero and then returned. Because the variance of $F_t$ (and thus of $M_t$ and $\mathbf{e}_t$) grows to infinity as $t$ tends to infinity, there is always a small chance of an extremely large fluxuation taking $\theta$ far away from zero. Off-policy TD(0), on the other hand, diverged to infinity in all individual runs.

For comparison, Figure 3 (right) shows trajectories for a $\theta \rightarrow 2\theta$ problem in which $F_t$ and all the other variables and their variances are bounded. In this problem, the target policy of selecting right on all steps leads to a soft terminal state ($\gamma(s) = 0$) with fixed value zero, which then transitions back to start again in the leftmost state, as shown in the upper right of the figure. (This is an example of how one can reproduce the conventional notions of terminal state and episode in a soft termination setting.) Here we have chosen the behavior policy to take the action left with probability 0.9, so that its stationary distribution distinctly favors the left state, whereas the target policy would spend equal time in each of the two states. This change increases the variance of the updates, so we used a smaller step size, $\alpha = 0.0001$; other settings were unchanged. Conventional off-policy TD(0) still diverged in this case, but emphatic TD(0) converged reliably to zero.

Finally, Figure 4 shows trajectories for the 5-state example shown earlier (and again in the upper part of the figure). In this case, everything is bounded under the target policy, and both algorithms converged. The emphatic algorithm achieved a lower MSVE in

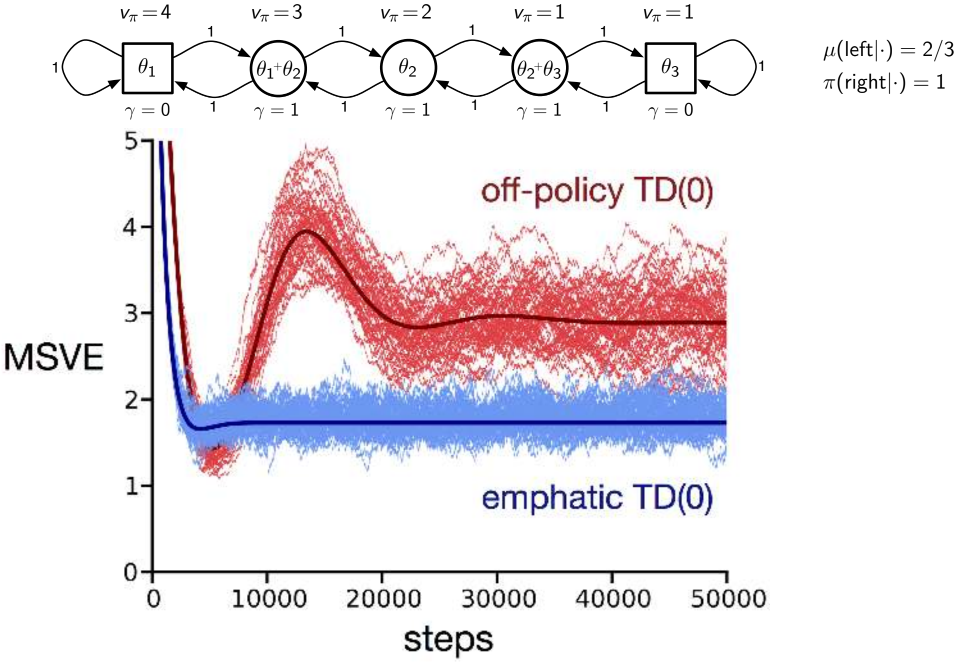


Figure 4: Twenty learning curves and their analytic expectation on the 5-state problem from Section 5, in which excursions terminate promptly and both algorithms converge reliably. Here $\lambda = 0$, $\boldsymbol{\theta}_0 = \mathbf{0}$, $\alpha = 0.001$, and $i(s) = 1, \forall s$. The MSVE performance measure is defined in (15).

this example (nevertheless, we do not mean to claim any general empirical advantage for emphatic TD($\lambda$) at this time).

Also shown in these figures as a thick dark line is the trajectory of the deterministic algorithm: $\bar{\boldsymbol{\theta}}_{t+1} = \bar{\boldsymbol{\theta}}_t + \alpha(\mathbf{b} - \mathbf{A}\bar{\boldsymbol{\theta}}_t)$ (6). Tsitsiklis and Van Roy (1997) argued that, for small step-size parameters and in the steady-state distribution, on-policy TD($\lambda$) follows its expected-update algorithm in an "average" sense, and we see much the same here for emphatic TD($\lambda$).

These examples show that although emphatic TD($\lambda$) is stable for any MDP and all functions $\lambda$, $\gamma$ and (positive) $i$, for some problems and functions the parameter vector continues to fluxuate with a chance of arbitrarily large deviations (for constant $\alpha > 0$). It is not clear how great of a problem this is. Certainly it is much less of a problem than the positive instability (Baird 1995) that can occur with off-policy TD($\lambda$) (stability of the expected update precludes this). The possibility of large fluxuations may be inherent in any algorithm for off-policy learning using importance sampling with long eligibility traces. For example, the updates of GTD($\lambda$) and GQ($\lambda$) (Maei 2011) with $\lambda = 1$ will tend to infinite variance as $t \to \infty$ on Baird's counterexample and on the example in Figures 1 and 3(left). And, as mentioned earlier, convergence with probability one can still be guaranteed if $\alpha$ is reduced appropriately (Yu 2015a,b).

In practice, however, even when asymptotic convergence can be guaranteed, high variance can be problematic as it may require very small step sizes and slow learning. High variance frequently arises in off-policy algorithms when they are Monte Carlo algorithms (no TD learning) or they have eligibility traces with high $\lambda$ (at $\lambda = 1$, TD algorithms become Monte Carlo algorithms). In both cases the root problem is the same: importance sampling ratios that become very large when multiplied together. For example, in the $\theta \rightarrow 2\theta$ problem discussed at the beginning of this section, the ratio was only two, but the products of successive twos rapidly produced a very large $F_t$. Thus, the first way in which variance can be controlled is to ensure that large products cannot occur. We are actually concerned with products of both $\rho_t$s and $\gamma_t$s. Occasional termination ($\gamma_t = 0$), as in the 5-state problem, is thus one reliable way of preventing high variance. Another is through choice of the target and behavior policies that together determine the importance sampling ratios. For example, one could define the target policy to be equal to the behavior policy whenever the followon or eligibility traces exceed some threshold. These tricks can also be done prospectively. White (personal communication) has proposed that the learner compute at each step the variance of what GTD($\lambda$)'s traces would be on the following step. If the variance is in danger of becoming too large, then $\lambda_t$ is reduced for that step to prevent it. For emphatic TD($\lambda$), the same conditions could be used to adjust $\gamma_t$ or one of the policies to prevent the variance from growing too large. Another idea for reducing variance is to use *weighted* importance sampling (as suggested by Precup et al. 2001) together with the ideas of Mahmood et al. (2014, 2015a) for extending weighted importance sampling to linear function approximation. Finally, a good solution may even be found by something as simple as bounding the values of $F_t$ or $\mathbf{e}_t$. This would limit variance at the cost of bias, which might be a good tradeoff if done properly (e.g., see Hallak, Tamar, Munos & Mannor 2015).

## 9. Conclusions and Future Work

We have introduced a way of varying the emphasis or strength of the updates of TD learning algorithms from step to step, based on importance sampling, that should result in much lower variance than previous methods (Precup et al. 2001). In particular, we have introduced the emphatic TD($\lambda$) algorithm and shown that it solves the problem of instability that plagues conventional TD($\lambda$) when applied in off-policy training situations in conjunction with linear function approximation. Compared to gradient-TD methods, emphatic TD($\lambda$) is simpler in that it has a single parameter vector and a single step size rather than two of each. The per-time-step complexities of gradient-TD and emphatic-TD methods are both linear in the number of parameters; both are much simpler than quadratic complexity methods such LSTD($\lambda$) and its off-policy variants. We have also presented a few empirical examples of emphatic TD(0) compared to conventional TD(0) adapted to off-policy training. These examples illustrate some of emphatic TD($\lambda$)'s basic strengths and weaknesses, but a proper empirical comparison with other methods remains for future work. Extensions of the emphasis idea to action-value and control methods such as Sarsa($\lambda$) and Q($\lambda$), to true-online forms (van Seijen & Sutton 2014, Sutton 2015), and to weighted importance sampling (Mahmood et al. 2014, 2015a) are also natural and remain for future work.

Yu (2015a) has recently extended the emphatic idea to a least-squares algorithm and proved that it and our emphatic TD($\lambda$) are convergent with probability one. She has also obtained a stronger result that does not require that the interest be positive in all states, using an argument similar to that given here about column sums and the Varga (1962) theorem for irreducibly diagonally dominant matrices (Yu 2015a, see also Mahmood et al. 2015b). Yu (2015b) proves convergence for weaker conditions on the step size. Asymptotic bounds on the error of emphatic TD($\lambda$) have been obtained by Hallak et al. (2015).

Two additional ideas for future work deserve special mention.

First, note that the present work has focused on ways of ensuring that the key matrix is positive definite, which implies positive definiteness of the **A** matrix and thus that the update is stable. An alternative strategy would be to work directly with the **A** matrix. Recall that the **A** matrix is vastly smaller than the key matrix; it has a row and column for each *feature*, whereas the key matrix has a row and column for each *state*. It might be feasible then to keep statistics for each row and column of **A**, whereas of course it would not be feasible for the large key matrix. For example, one might try to use such statistics to directly test for diagonal dominance (and thus positive definiteness) of **A**. If it were possible to adjust some of the free parameters (e.g., the $\lambda$ or $i$ functions) to ensure positive definiteness while reducing the variance of $F_t$, then a substantially improved algorithm might be found.

The second idea for future work is that the emphasis algorithm, by tracing the dependencies among the estimates at various states, is doing something clever that ought to show up as improved bounds on the asymptotic approximation error. The bound given by Tsitsiklis and Van Roy (1997) probably cannot be significantly improved if $\lambda$, $\gamma$, $i$, and $\rho$ are all constant, because in this case emphasis asymptotes to a constant that can be absorbed into the step size. But if any of these vary from step to step, then emphatic TD($\lambda$) is genuinely different and may improve over conventional TD($\lambda$). In particular, consider an episodic on-policy case where $i(s) \doteq 1$ and $\lambda(s) \doteq 0$, for all $s \in \mathcal{S}$, and $\gamma(s) \doteq 1$ for all states except for a terminal state where it is zero (and from which a new episode starts). In this case emphasis would increase linearly within an episode to a maximum on the final state, whereas conventional TD($\lambda$) would give equal weight to all steps within the episode. If the feature representation were insufficient to represent the value function exactly, then the emphatic algorithm might improve over the conventional algorithm in terms of asymptotic MSVE (15). Similarly, improvements in asymptotic MSVE over conventional algorithms might be possible whenever $i$ varies from state to state, such as in the common episodic case in which we are interested only in accurately valuing the start state of the episode, and yet we choose $\lambda < 1$ to reduce variance. There may be a wide range of interesting theoretical and empirical work to be done along these lines.

## Acknowledgements

The authors thank Hado van Hasselt, Doina Precup, Huizhen Yu, Susan Murphy, and Brendan Bennett for insights and discussions contributing to the results presented in this paper, and the entire Reinforcement Learning and Artificial Intelligence research group for providing the environment to nurture and support this research. We gratefully acknowledge funding from Alberta Innovates – Technology Futures and from the Natural Sciences and Engineering Research Council of Canada.

## References

Baird, L. C. (1995). Residual algorithms: Reinforcement learning with function approximation. *International Conference on Machine Learning 12*, pp. 30–37. Morgan Kaufmann. Important modifications and errata added to the online version on November 22, 1995.

Bertsekas, D. P. (2012). *Dynamic Programming and Optimal Control: Approximate Dynamic Programming*, Fourth Edition. Athena Scientific, Belmont, MA.

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

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

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

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

Dann, C., Neumann, G., Peters, J. (2014). Policy evaluation with temporal differences: A survey and comparison. *Journal of Machine Learning Research 15*:809–883.

Geist, M., Scherrer, B. (2014). Off-policy learning with eligibility traces: A survey. *Journal of Machine Learning Research 15*:289–333.

Gordon, G. J. (1995). Stable function approximation in dynamic programming. *International Conference on Machine Learning 12*, pp. 261–268. An expanded version was published as Technical Report CMU-CS-95-103, Carnegie Mellon University, Pittsburgh, PA, 1995.

Gordon, G. J. (1996). Stable fitted reinforcement learning. *Advances in Neural Information Processing Systems 8*, pp. 1052–1058.

Hackman, L. (2012). *Faster Gradient-TD Algorithms.* MSc thesis, University of Alberta.

Hallak, A., Tamar, A., Munos, R., Mannor, S. (2015). Generalized emphatic temporal difference learning: Bias-variance analysis. ArXiv:1509.05172.

Klopf, A. H. (1988). A neuronal model of classical conditioning. *Psychobiology 16*(2):85–125.

Kolter, J. Z. (2011). The fixed points of off-policy TD. *Advances in Neural Information Processing Systems 24*, pp. 2169–2177.

Lagoudakis, M., Parr, R. (2003). Least squares policy iteration. *Journal of Machine Learning Research 4*:1107–1149.

Ludvig, E. A., Sutton, R. S., Kehoe, E. J. (2012). Evaluating the TD model of classical conditioning. *Learning & behavior 40*(3):305–319.

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

Maei, H. R., Sutton, R. S. (2010). GQ($\lambda$): A general gradient algorithm for temporal-difference prediction learning with eligibility traces. *Artificial General Intelligence 3*, pp. 91–96.

Maei, H. R., Szepesvári, Cs., Bhatnagar, S., Sutton, R. S. (2010). Toward off-policy learning control with function approximation. *International Conference on Machine Learning 27*, pp. 719–726.

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. (2015a). Off-policy learning based on weighted importance sampling with linear computational complexity. *Uncertainty in Artificial Intelligence 31.*

Mahmood, A. R., Yu, H., White, M., Sutton, R. S. (2015b). Emphatic temporal-difference learning. *European Workshop on Reinforcement Learning 12*, ArXiv:1507.01569.

Modayil, J., White, A., Sutton, R. S. (2014). Multi-timescale nexting in a reinforcement learning robot. *Adaptive Behavior 22*(2):146–160.

Nedić, A., Bertsekas, D. P. (2003). Least squares policy evaluation algorithms with linear function approximation. *Discrete Event Dynamic Systems 13*(1-2):79–110.

Niv, Y., Schoenbaum, G. (2008). Dialogues on prediction errors. *Trends in cognitive sciences 12*(7):265–272.

O'Doherty, J. P. (2012). Beyond simple reinforcement learning: The computational neurobiology of reward learning and valuation. *European Journal of Neuroscience 35*(7):987–990.

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

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

Rummery, G. A. (1995). *Problem Solving with Reinforcement Learning.* PhD thesis, University of Cambridge.

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal on Research and Development 3*:210–229. Reprinted in E. A. Feigenbaum & J. Feldman (Eds.), *Computers and thought*, McGraw-Hill, New York.

Schultz, W., Dayan, P., Montague, P. R. (1997). A neural substrate of prediction and reward. *Science 275*(5306):1593–1599.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning 3*:9–44, erratum p. 377.

Sutton, R. S. (1995). TD models: Modeling the world at a mixture of time scales. *International Conference on Machine Learning 12*, pp. 531–539.

Sutton, R. S. (2009). The grand challenge of predictive empirical abstract knowledge. *Working Notes of the IJCAI-09 Workshop on Grand Challenges for Reasoning from Experiences.*

Sutton, R. S. (2012). Beyond reward: The problem of knowledge and data. *International Conference on Inductive Logic Programming 21*, S. H. Muggleton, A. Tamaddoni-Nezhad, F. A. Lisi (Eds.): ILP 2011, LNAI 7207, pp. 2–6. Springer, Heidelberg.

Sutton, R. S. (2015). True online emphatic TD($\lambda$): Quick reference and implementation guide. ArXiv 1507.07147.

Sutton, R. S., Barto, A. G. (1990). Time-derivative models of Pavlovian reinforcement. In M. Gabriel and J. Moore (Eds.), *Learning and Computational Neuroscience: Foundations of Adaptive Networks*, pp. 497–537. MIT Press, Cambridge, MA.

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

Sutton, R. S., Mahmood, A. R., Precup, D., van Hasselt, H. (2014). A new Q($\lambda$) with interim forward view and Monte Carlo equivalence. *International Conference on Machine Learning 31. JMLR W&CP 32*(2).

Sutton, R. S., Maei, H. R., Precup, D., Bhatnagar, S., Silver, D., Szepesvári, Cs., Wiewiora, E. (2009). Fast gradient-descent methods for temporal-difference learning with linear function approximation. *International Conference on Machine Learning 26*, pp. 993–1000.

Sutton, R. S., Modayil, J., Delp, M., Degris, T., Pilarski, P. M., White, A., Precup, D. (2011). Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. *International Conference on Autonomous Agents and Multiagent Systems 10*, pp. 761–768.

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

Sutton, R. S., Rafols, E. J., Koop, A. (2006). Temporal abstraction in temporal-difference networks. *Advances in Neural Information Processing Systems 18.*

Tesauro, G. (1992). Practical issues in temporal difference learning. *Machine Learning 8*:257–277.

Tesauro, G. (1995). Temporal difference learning and TD-Gammon. *Communications of the ACM 38*(3):58–68.

Thomas, P. (2014). Bias in natural actor–critic algorithms. *International Conference on Machine Learning 31. 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.

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

van Seijen, H., Sutton, R. S. (2014). True online TD($\lambda$). *International Conference on Machine Learning 31. JMLR W&CP 32*(1):692–700.

Varga, R. S. (1962). *Matrix Iterative Analysis.* Englewood Cliffs, NJ: Prentice-Hall.

Wang, M., Bertsekas, D. P. (2013). Stabilization of stochastic iterative methods for singular and nearly singular linear systems. *Mathematics of Operations Research 39*(1):1-30.

Watkins, C. J. C. H. (1989). *Learning from Delayed Rewards.* PhD thesis, University of Cambridge.

Watkins, C. J. C. H., Dayan, P. (1992). Q-learning. *Machine Learning 8*:279–292.

White, A. (2015). *Developing a Predictive Approach to Knowledge.* Phd thesis, University of Alberta.

Yu, H. (2010). Convergence of least squares temporal difference methods under general conditions. *International Conference on Machine Learning 27*, pp. 1207–1214.

Yu, H. (2012). Least squares temporal difference methods: An analysis under general conditions. *SIAM Journal on Control and Optimization 50*(6):3310–3343.

Yu, H. (2015a). On convergence of emphatic temporal-difference learning. ArXiv:1506.02582. A shorter version appeared in *Conference on Learning Theory 18*, *JMLR W&CP 40*.

Yu, H. (2015b). Weak convergence properties of constrained emphatic temporal-difference learning with constant and slowly diminishing stepsize. ArXiv:1511.07471.