A First Empirical Study of Emphatic Temporal Difference Learning

Sina Ghiassian
Department of Computing Science
University of Alberta
ghiassia@ualberta.ca

Banafsheh Rafiee
Department of Computing Science
University of Alberta
rafiee@ualberta.ca

Richard S. Sutton
Department of Computing Science
University of Alberta
rsutton@ualberta.ca

Abstract

In this paper we present the first empirical study of the emphatic temporal-difference learning algorithm (ETD), comparing it with conventional temporal-difference learning, in particular, with linear TD(0), on on-policy and off-policy variations of the Mountain Car problem. The initial motivation for developing ETD was that it has good convergence properties under *off*-policy training (Sutton, Mahmood & White 2016), but it is also a new algorithm for the *on*-policy case. In both our on-policy and off-policy experiments, we found that each method converged to a characteristic asymptotic level of error, with ETD better than TD(0). TD(0) achieved a still lower error level temporarily before falling back to its higher asymptote, whereas ETD never showed this kind of "bounce". In the off-policy case (in which TD(0) is not guaranteed to converge), ETD was significantly slower.

1 Emphatic Temporal Difference Learning

We consider the problem of learning the value function for a Markov decision process and a given policy. An agent and environment interact at discrete time steps, $t = 0, 1, 2, \ldots$, at each of which the environment is in a state S_t, the agent selects an action A_t and as a result the environment emits a reward R_{t+1} and a next state S_{t+1}. States are represented to the agent as feature vectors $\phi_t = \phi(S_t) \in \mathbb{R}^n$. We seek to find a parameter vector, $\theta_t \in \mathbb{R}^n$ such that the inner product $\theta_t^\top \phi_t$ approximates the expected return $\mathbb{E}\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots \mid A_{t:\infty} \sim \pi\right]$, where $\pi : \mathcal{A} \times \mathcal{S} \to [0, 1]$ is a policy for selecting the future actions. In fact, all actions are selected by an alternate policy μ. If $\pi = \mu$, then the training is called *on-policy*, whereas if the two policies are different the training is called *off-policy*.

We consider the special case of the emphatic temporal difference learning algorithm (ETD) in which bootstrapping is complete ($\lambda(s) = 0, \forall s$) and there is no discounting ($\gamma(s) = 1, \forall s$). Studying TD and ETD methods with complete bootstrapping is suitable because in this case the differences between them are maximized. As λ approaches 1, the methods behave more similarly up to the point where they become equivalent when $\lambda = 1$. By setting $\lambda = 0$ and $\gamma = 1$, the ETD algorithm can be

completely described by:

$$\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha \rho_t F_t \left(R_{t+1} + \boldsymbol{\theta}_t^T \boldsymbol{\phi}_{t+1} - \boldsymbol{\theta}_t^T \boldsymbol{\phi}_t \right) \boldsymbol{\phi}_t,$$
$$F_t \doteq \rho_{t-1} F_{t-1} + 1, \text{ with } F_0 \doteq 1,$$
$$\rho_t \doteq \frac{\pi(A_t|S_t)}{\mu(A_t|S_t)},$$

where $\alpha > 0$ is a step size parameter. F is the followon trace according to which the update at each time step is emphasized or de-emphasized. TD is obtained by removing the F from the first equation. Because of F, ETD is different from TD even in the on-policy case in which ρ is always 1. For a thorough explanation of ETD see (Sutton, Mahmood & White 2016).

2   Stability of On-policy TD with Variable λ: A Counterexample

In this section we show that although the initial motivation for developing ETD was that it has good convergence properties under off-policy training (Yu 2015), it is also a different algorithm under on-policy training. To emphasize the difference between the two, we present a simple example for which TD(λ) is *not convergent* under *on-policy* training but ETD is.

It has long been known that TD(λ) converges with any constant value of λ under on-policy training (Tsitsiklis & Van Roy 1997). Surprisingly, TD(λ) is not assured to converge with varying λ even under on-policy training. Yu has recently presented a counterexample (personal communication) with state dependent λ for which on-policy TD(λ) is not convergent. The example is a simple Markov decision process consisting of two states in which the system simply moves from one state to another in a cycle. The process starts in each of the states with equal probability. Let $\lambda(S_1) = 0$ and $\lambda(S_2) = 1$, $\boldsymbol{\phi}(S_1) = (3, 1)$ and $\boldsymbol{\phi}(S_2) = (1, 1)$ and $\gamma = 0.95$. As shown below, the TD(λ) key matrix for this problem is not positive definite. Moreover, both eigenvalues of the key matrix have negative real parts and thus TD(λ) diverges in this case.

$$\text{Key matrix} = \begin{pmatrix} -0.4862 & 0.1713 \\ -0.7787 & 0.0738 \end{pmatrix}$$
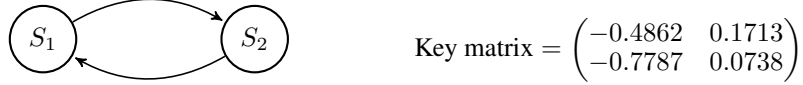
This is while ETD is convergent under both off-policy and on-policy training with variable λ. This example appears in more detail in the supplementary material.

3   Fixed-policy Mountain Car Testbed

For our experimental study, we used a new variation of the mountain car control problem (Sutton & Barto 1998) to form a prediction problem. The original mountain car problem has a 2-dimensional space, position (between -1.2 and 0.6), and velocity (between -0.07 and 0.07) with three actions, full throttle forward, full throttle backward, and 0 throttle. Each episode starts around the bottom of a hill (a uniform random number between -0.4 and -0.6). The reward is -1 on all time steps until the car pasts its goal at the top of the hill, which ends the episode. The task is undiscounted. Our variation of the mountain car problem has a fixed target policy which is to always push towards the direction of the velocity and not to push in any direction when the velocity is 0. We call the new variation of the mountain car problem, the fixed-policy mountain car testbed.

The performance measure we used is an estimation of the mean squared value error (MSVE) which reflects the mean squared difference between the true value function and the estimated value function, weighted according to how often each state is visited in the state space following the behavior policy:

$$\widehat{MSVE}(\boldsymbol{\theta}) = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} [\hat{v}(s, \boldsymbol{\theta}) - v_\pi(s)]^2$$

$\mathcal{S}$ included 500 sample states gathered by following the behaviour policy for 10,000,000 steps and randomly choosing 500 states from the last 5,000,000. We did not use the first 5,000,000 because

the state distribution may change as more steps are taken and the stationary distribution is achieved in the limit. The agent started from each state $s \in \mathcal{S}$ and followed the target policy to termination 1,000 times, each time the return was computed and recorded. All 1,000 returns were averaged and the result was used as the true value of the state value function, $v_\pi(s)$. The learning algorithm's estimation of the value function for state s is shown by $\hat{v}(s, \boldsymbol{\theta}) = \boldsymbol{\theta}^\top \boldsymbol{\phi}(s)$.

4   On-Policy Experiments

We applied on-policy TD and on-policy ETD methods to the fixed-policy mountain car testbed. We created many instances of each method by changing the step size parameter. To approximate a value function for this problem, we used tile coding (Sutton 1996) with 5 tilings, 4×4 tiles each. Each algorithm instance was initialized with a 0 weight vector, and then run for 500,000 episodes. The whole process was repeated for 50 runs.

To produce learning curves for each instance of the two methods we computed the error measure at the end of each episode and averaged over runs. See Figure 1a. We also performed a parameter study of the asymptotic performance for both methods. To do so, we averaged the error of the last 1% of the episodes for each run, and then computed the average and standard error over all 50 runs. See Figure 1b.
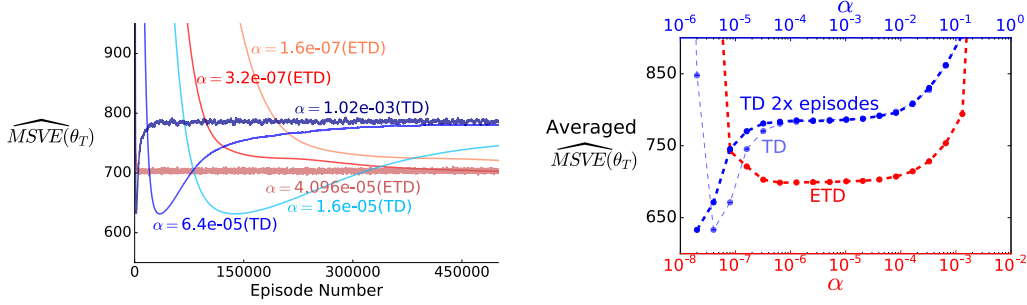
(a) **Learning curves**: $\widehat{MSVE}$ was computed after each episode when the terminal state (T) was reached and thus is termed $\widehat{MSVE}(\boldsymbol{\theta}_T)$.

(b) **Parameter study**: $\widehat{MSVE}$ was computed after each episode when the terminal state (T) was reached and then was averaged over the last 1% episodes. This was done for each different value of α. The standard errors of different instances are shown as error bars on the figure but are not visible due to their small sizes.

Figure 1: Results of on-policy ETD and TD methods on the fixed-policy mountain car testbed

To compare the performance of on-policy TD and on-policy ETD, we first need to understand how their errors changed as the number of episodes increased. ETD's error was a decreasing function of the number of episodes for sufficiently small values of α. However, TD showed a bounce, reaching a low error temporarily before falling back to its higher asymptotic error. The depth and the asymptotic level of the bounce did not depend on α, but its duration did. The smaller the α, the later the bounce and as a result, it took more than 500,000 episodes for TD to converge for smaller values of α. See Figure 1a.

ETD outperformed TD in terms of asymptotic performance. TD instances with smaller values of the step size ($\alpha < 10^{-4}$) did not converge within 500,000 episodes. See Figure 1b. To confirm that TD has not converged for smaller values of α, we repeated the TD experiments for 1,000,000 episodes and computed the error measure. The error measure changed *only* for the instances that did not converged within 500,000 episodes. The light and the dark blue curves in Figure 1b show the performance of different instances of the TD method after 500,000 and 1,000,000 episodes respectively. It is obvious that TD instances with $\alpha < 10^{-4}$ did not converge while the instances with larger values of the step size did.

5 Off-Policy Experiments

We also applied off-policy TD and off-policy ETD to the fixed-policy mountain car testbed. In this case, the target policy was the same as the policy in the on-policy case and the behavior policy was to choose a random action 10% of the time and act according to the target policy 90% of the time. Again different instances of each method was created with different step size parameters. Each instance of the method was run for 500,000 episodes and the whole process was repeated for 50 runs. The learning curves for the off-policy case are presented in Figure 2a. The parameter study results are in Figure 2b.
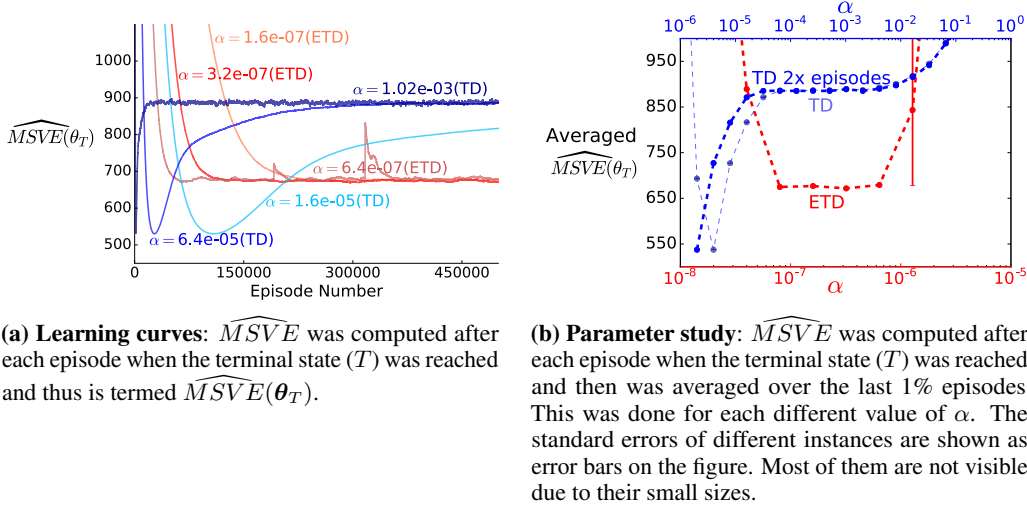

(a) Learning curves: $\widehat{MSVE}$ was computed after each episode when the terminal state (T) was reached and thus is termed $\widehat{MSVE}(\boldsymbol{\theta}_T)$.

(b) Parameter study: $\widehat{MSVE}$ was computed after each episode when the terminal state (T) was reached and then was averaged over the last 1% episodes. This was done for each different value of α. The standard errors of different instances are shown as error bars on the figure. Most of them are not visible due to their small sizes.

Figure 2: Results of off-policy ETD and TD methods on the fixed-policy mountain car testbed

Analogous to the on-policy case, each method had its advantages and disadvantages. ETD achieved a better asymptotic performance whenever it converged. This is while TD, compared to ETD, could take advantage of using larger values of step size and thus converged significantly faster (Figure 2a). ETD's step size values had to be set small (in the order of 10^{-7}) to control the method's high variance (Figure 2b). TD had a larger step size range with which it converged; however, ETD converged only for a short range of step size (Figure 2b). Similar to the on-policy study, TD showed a bounce for every value of step size while ETD did not (Figure 2a).

6 Conclusion

We performed the first systematic empirical study of the emphatic temporal difference learning method and showed that it can be used in a problem with a relatively large state space with promising results. Although ETD is originally proposed as an off-policy method, it can also be used as a reliable on-policy algorithm. According to our results, ETD seems to be slow in the off-policy case; however, it achieves a better asymptotic performance in both on-policy and off-policy cases. In spite of the fact that our experiments are limited to a variation of the mountain car problem, we believe that our observations can lead to a better understanding of both TD and ETD methods. Yu's counter example along with our experimental results motivate further study of ETD as an on-policy or off-policy method.

References

Sutton, R.S. (1988). Learning to Predict by the Methods of Temporal Differences. Machine learning 3(1):9-44.

Sutton, R .S. (1996). Generalization in Reinforcement Learning: Successful Examples Using Sparse Coarse Coding. Advances in Neural Information Processing Systems:1038-1044.

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

Sutton, R. S., Mahmood, A. R., White, M. (2015). An Emphatic Approach to the Problem of Off-policy Temporal-difference Learning. The Journal of Machine Learning Research.

Tsitsiklis, J .N., Van Roy, B. (1997). An Analysis of Temporal-difference Learning with Function Approximation. IEEE Transactions on Automatic Control 42(5):674-690.

Yu, H. (2015). On convergence of emphatic temporal-difference learning. In Proceedings of the Conference on Computational Learning Theory.

7 Supplementary Material

7.1 Stability of On-policy TD with Variable λ: A Counterexample

Suppose the policy induces an irreducible Markov chain with transition matrix P_π and a unique invariant probability distribution μ (i.e., $\mu^\top P_\pi = \mu^\top$). Let $D = diag(\mu)$ and let Φ be a feature matrix with linearly independent columns. The key matrix associated with the TD(λ) algorithm is $A = \Phi^\top D(I - P_\pi^\lambda)\Phi$, where P_π^λ is a substochastic matrix determined by P_π, λ and the discount factor γ. For a constant $\lambda \in [0, 1]$, the matrix A is positive definite (see e.g., Tsitsiklis and Van Roy 1997), ensuring the stability of the algorithm. This positive definiteness property relies critically on the fact that $\mu^\top P_\pi^\lambda < \mu^\top$, which does not hold in general when λ is a function of states. Thus, with state-dependent λ, the positive definiteness of the matrix A and the stability of the TD(λ) algorithm are no longer guaranteed.

In our example $\lambda(S_1) = 0$, $\lambda(S_2) = 1$, $\mu^\top = (0.5, 0.5)$ and $P_\pi^\lambda = \begin{pmatrix} \gamma^2 & 0 \\ \gamma & 0 \end{pmatrix}$. For γ near 1, e.g., $\gamma = 0.95$, and for Φ as given below, we can calculate the matrix A associated with TD(λ):

$$\Phi = \begin{pmatrix} 3 & 1 \\ 1 & 1 \end{pmatrix}, \qquad A = \Phi^\top D(I - P_\pi^\lambda)\Phi = \begin{pmatrix} -0.4862 & 0.1713 \\ -0.7787 & 0.0738 \end{pmatrix}.$$

The matrix A is not positive definite. Moreover, both eigenvalues of A have negative real parts, and hence $-A$ is not a Hurwitz matrix and TD(λ) diverges in general in this case.