Step-size Adaptation for TD(λ) — Comparing Two Algorithms

Khurram Javed[*]

July 25, 2021

Algorithm 1: TIDBD*(λ) and TIDBD(λ)

```
Initialize $h_i$ and $z_i$ to 0 and $\theta$, $w_i$, and $\beta_i$ as desired for $i \in \{1, 2, \cdots, n\}$;
Observe state features $\phi(s)$;
for each observation $\phi(s')$ and R;
 do
    $\delta = R + \gamma \sum_{i=1}^{n} w_i \phi({s'}_i) - \sum_{i=1}^{n} w_i \phi(s_i)$;
    for $i = 1, 2, \cdots, n$;
     do
        $\beta_i = \beta_i + \theta \delta z_i h_i$ // Update proposed by Thill (2015)
        $\beta_i = \beta_i + \theta \delta \phi_i(s) h_i$ // Update proposed by TIDBD(λ)
        $a_i = e^{\beta_i}$;
        $z_i = z_i \gamma \lambda + \phi_i(s)$;
        $w_i = w_i + \alpha_i \delta z_i$;
        $h_i = h_i[1 - \alpha_i \phi(s)_i z_i]^+ + \alpha_i \delta z_i]$;
    end
end
```

Incremental-Delta-Bar-Delta (IDBD) is an online step-size adaptation algorithm for linear regression (Sutton, 1992). It has been extended to learn step-sizes for the TD(λ) algorithm (Sutton und Barto, 2018) in two independent works. Thill (2015) extended a non-linear version of IDBD — linear weighted sum of features followed by a sigmoid activation — for TD(λ), whereas Kearney u. a. (2018) extended the linear version of IDBD for TD(λ) — called TIDBD(λ). The two algorithms optimize a different meta-objective for learning the step-sizes. I remove the non-linearity from the algorithm proposed by Thill (2015) and call it TIDBD*(λ) in this document. I compare TIDBD(λ) and TIDBD*(λ).

TIDBD(λ) meta-learns the step-sizes by minimizing the TD(0) objective and learns the parameters by minimizing the TD(λ) objective. TIDBD*(λ), on the other hand, minimizes the TD(λ) objective to learn both the weights and the

[*]RLAI Lab, University of Alberta. Correspondence at: kjaved@ualberta.ca

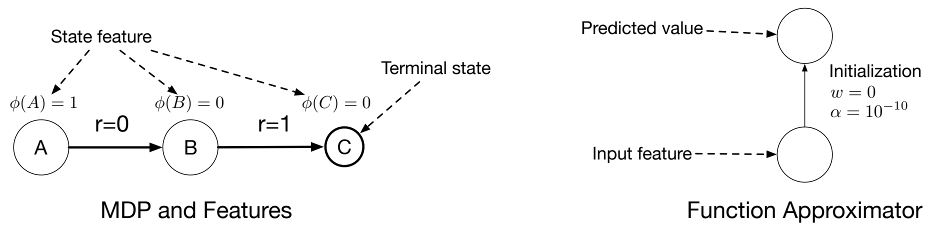

Figure 1: The three state MDP (left) and the value network with a scalar weight w initialized to be zero (right). The learner uses TD(λ) with $\lambda = 1$ and $\gamma = 1$. Because the feature associated with the second and third state is 0, the learner can only predict zero as the value. However for the first state, it can learn to output the correct value — one. The step-size is initialized to be a very small value of 10^{-10}. The optimal solution for this problem for $\lambda = 1$ and $\gamma = 1$ is $w = 1$.

step-sizes. The two algorithms behave differently for problems for which TD(0) and TD(λ) learn differently.

1 Empirical Demonstration

I elucidate the difference between the two algorithms by learning a scalar weight on a simple MDP for which the optimal parameter w.r.t TD(0) objective is zero and that w.r.t the TD(1) objective is one. The MDP has three states — A, B, and C. Every episode starts at state A, goes to B, and then terminates at C. Reward from A to B and B to C is zero and one, respectively. Each state has a scalar feature. State A, B, and C have features one, zero, and zero, respectively. The function approximator uses the feature with a scalar weight to predict the value. Figure 1 shows both the MDP and the function approximator. Before learning, the scalar weight of the function approximator is zero, and its step-size is 10^{-10}. Both λ and γ are one.

I run both TIDBD(λ) (Kearney u. a., 2018) and TIDBD*(λ) (Thill, 2015) for five million steps and report the results in Figure 2. I use a meta step-size — θ — of one and reset the trace z to zero at the beginning of every episode. TIDBD(λ) does not increase the step-size of w and as a result, does not converge to the optimal weight in five million steps. This behavior is not surprising — TIDBD(λ) is computing the meta-gradient w.r.t the TD(0) objective and because $w = 0$ is already the optimal value, it sees no reason to increase the step-size. TIDBD*(λ), on the other hand, increases the step-size until w reaches one. Then it slowly reduces the step-size, converging to $w = 1$.

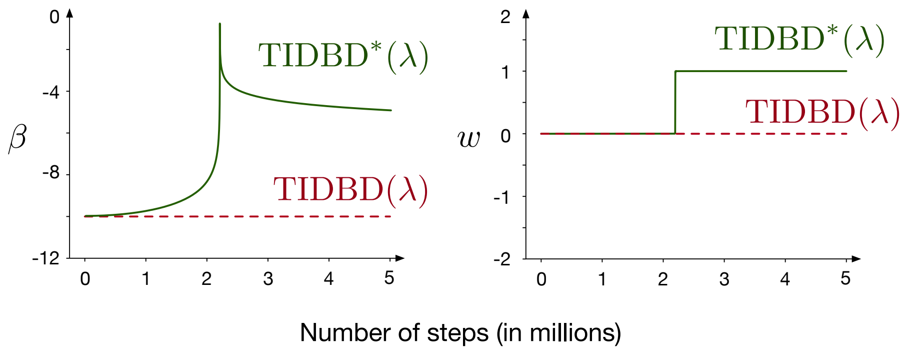

Figure 2: Results of TIDBD(λ) and TIDBD*(λ) on the three state MDP. TIDBD*(λ) increases the step-size and converges to the optimal weight. TIDBD(λ), on the other hand, never increases the step-sizes and as a result does not converge to the optimal weight with-in 5 million steps.

2 Conclusion

My recommendation is to use TIDBD*(λ), proposed by Thill (2015), when doing step-size adaptation with eligibility traces. The requires a single line change from TIDBD(λ), as shown in Algorithm 1.

References

[Kearney u. a. 2018] KEARNEY, Alex ; VEERIAH, Vivek ; TRAVNIK, Jaden B. ; SUTTON, Richard S. ; PILARSKI, Patrick M.: Tidbd: Adapting temporal-difference step-sizes through stochastic meta-descent. In: *arXiv preprint arXiv:1804.03334* (2018)

[Sutton 1992] SUTTON, Richard S.: Adapting bias by gradient descent: An incremental version of delta-bar-delta. In: *AAAI*, 1992

[Sutton und Barto 2018] SUTTON, Richard S. ; BARTO, Andrew G.: *Reinforcement learning: An introduction*. MIT press, 2018

[Thill 2015] THILL, Markus: *Temporal difference learning methods with automatic step-size adaption for strategic board games: Connect-4 and Dots-and-Boxes*, Master thesis, Cologne University of Applied Sciences, June 2015, Dissertation, 2015