Online TD(1) Meets Offline Monte Carlo

Yao-Liang Yu YAOLIANG@CS.UALBERTA.CA
Yongjian Zhang YONGJIAN@CS.UALBERTA.CA
Csaba Szepesvári SZEPESVA@CS.UALBERTA.CA
Department of Computing Science, University of Alberta, Edmonton, AB T6G 2E8 Canada

Keywords: adaptive stepsize, online learning, offline learning, temporal difference, Monte Carlo

Abstract

We extend the recent adaptive temporal difference (ATD) learning (Hutter & Legg, 2008) to absorbing MDPs and demonstrate that in this case ATD is actually an online version of (*every-visit*) Monte Carlo updating.

1. Introduction

Temporal difference (TD) learning and Monte Carlo (MC) simulation are two major streams in reinforcement learning (RL) (Sutton & Barto, 1998), especially when a model is not available. Although it is still not clear if TD is *consistently* better than MC or vice versa, the latter has been accused of being only applicable in absorbing MDPs. The reason is "obvious": MC is an offline algorithm, while TD, an inherently online algorithm, could work in both absorbing and ergodic MDPs.

On the other hand, TD and MC are also closely connected by the eligibility trace, one of the most important techniques in RL (Sutton & Barto, 1998). The equivalence between offline TD(1) and MC has been established in (Singh & Sutton, 1996). Despite its theoretical importance, offline TD(1) is rarely used in practice since if we can wait until the end of episode (and then do backups), why not use MC directly? Instead, one usually favors online TD(1) and expects its performance to be *similar* to offline TD(1), provided that the stepsize is small. Then an interesting question to ask is does there exist a particular form of stepsize that *bridges* online TD(1) and offline TD(1) (or MC equivalently)?

We provide an affirmative answer to the above question. Our result is based on (Hutter & Legg, 2008), which we call adaptive temporal difference (ATD) learning. Note that the original ATD can *only* work on ergodic MDPs. We extend ATD to absorbing MDPs and show that an online TD(1) algorithm with adaptive stepsize does offline (*every-visit*) MC's job *exactly*! In other words, we find an online MC algorithm, which might be helpful for us to rethink the strengths and weaknesses of MC in RL.

We focus on tabular RL prediction problem, though we have some initial results on linear function approximation (quite similar to LSTD (Bradtke & Barto, 1996)). We are also aware of that it is possible to have equivalence between online and offline TD if the TD error is slightly changed (Sutton & Barto, 1998).

2. ATD in Absorbing MDPs

In order for ATD to be able to work in absorbing MDPs, we split its objective function into two parts:

$$L := \frac{1}{2}\Big[\sum_{k=1}^{T}\lambda^{t-k}(R_k - V_{s_k}^{(t)})^2 + \sum_{k=T+1}^{t}\lambda^{t-k}(\tilde{R}_k - V_{s_k}^{(t)})^2\Big] \tag{1}$$

$T+1$ is the starting time of the *current* episode, λ is used in non-stationary environments to discount previous experience. In the first term of (1), R_k (returns of *finished* episodes) are perfectly known to us, while in the second term, bootstrapping is needed to approximate $\tilde{R}_k$, which we don't know since the *current* episode is still ongoing.

We take the derivative of L with respect to $V_s^{(t)}$:

$$\frac{\partial L}{\partial V_s^{(t)}} = \sum_{k=1}^{T}\lambda^{t-k}(V_{s_k}^{(t)} - R_k)\mathbb{I}_{s_k s} + \sum_{k=T+1}^{t}\lambda^{t-k}(V_{s_k}^{(t)} - \tilde{R}_k)\mathbb{I}_{s_k s} \tag{2}$$

Here $\mathbb{I}$ is the indicator function: $\mathbb{I}_{s_k s} = 1$ **iff** $s_k = s$. We then bootstrap $\tilde{R}_k$ in the *current* ongoing episode with $R_k^{(t-k)}$, the so called n-step return (Sutton & Barto, 1998):

$$R_k^{(t-k)} := \sum_{u=k}^{t-1}\gamma^{u-k}r_u + \gamma^{t-k}V_{s_t}^{(t)} \tag{3}$$

Note that for different states which have been experienced in the current episode, we bootstrap them with n-step returns of different lengths. As will be pointed out in **Remark 2**, this step is critical to mimick Monte Carlo updating.

By setting the derivative (2) to zero and rearranging terms, we will get:

$$V_s^{(t)} N_s^{(t)} = M_s^{(t)} + C_s^{(t)} + E_s^{(t)} V_{s_t}^{(t)} \tag{4}$$

where $N_s^{(t)} := \sum_{k=1}^{t} \lambda^{t-k} \mathbb{I}_{s_k s}$, $C_s^{(t)} := \sum_{k=1}^{T} \lambda^{t-k} R_k \mathbb{I}_{s_k s}$, $E_s^{(t)} := \sum_{k=T+1}^{t} (\lambda\gamma)^{t-k} \mathbb{I}_{s_k s}$, and $M_s^{(t)} := \sum_{u=T+1}^{t-1} \lambda^{t-u} E_s^{(u)} r_u$

Remark 1. For ergodic MDPs, T is always zero and the first term in (1) will disappear. However, the above procedure still holds. In fact, this is the original ATD algorithm presented in (Hutter & Legg, 2008).

Remark 2. If the current episode happens to end at time t (say T'), we see that n-step return $R_k^{(t-k)}$ is exactly $\tilde{R}_k$ (no bootstrapping needed). Therefore, the second term in (1) will be absorbed into the first term and ATD is simplified to the following weighted linear regression problem:

$$\min \quad \frac{1}{2} \sum_{k=1}^{T'} \lambda^{T'-k} (R_k - V_{s_k}^{(T')})^2 \tag{5}$$

If we set $\lambda = 1$, then it is easy to see that (5) is every-visit MC sample average, which indicates that (4) will be exactly every-visit MC updating when the current episode terminates.

Instead of solving (4) each time step, we would like to have an online algorithm. This is possible by exploiting the following incremental update rules: $C_s^{(t+1)} = \lambda C_s^{(t)}$, $N_s^{(t+1)} = \lambda N_s^{(t)} + \mathbb{I}_{s_{t+1} s}$, $E_s^{(t+1)} = \lambda\gamma E_s^{(t)} + \mathbb{I}_{s_{t+1} s}$ and $M_s^{(t+1)} = \lambda M_s^{(t)} + \lambda E_s^{(t)} r_t$. We give the online update rule (details are omitted):

$$V_s^{(t+1)} = V_s^{(t)} + \alpha_t(s, s_{t+1}) \cdot E_s^{(t)} \cdot \delta_t \tag{6}$$

$$\alpha_t(s, s_{t+1}) := \frac{N_{s_{t+1}}^{(t)}}{N_{s_{t+1}}^{(t)} - \gamma E_{s_{t+1}}^{(t)}} \cdot \frac{1}{N_s^{(t)}} \tag{7}$$

where $\delta_t = r_t + \gamma V_{s_{t+1}}^{(t)} - V_{s_t}^{(t)}$ is the well-known TD error, and $E_s^{(t)}$ is the eligibility trace.

Remark 3. Note that (6) is very similar to the original TD algorithm (Sutton & Barto, 1998) except that the stepsize α is adaptive. Since (6)-(7) comes from (4), it is easy to see that if we set $\lambda = 1$, at the end of the episode, the (in-place) updated state value V in (6) exactly equals that of every-visit MC updating.

Remark 4. $N_s^{(t)}$ records how many times we have visited state s until time step t. The key point in extending (Hutter & Legg, 2008) to absorbing MDPs is that we *never* clear $N_s^{(t)}$ throughout episodes while eligibility trace $E_s^{(t)}$ is as usual cleared after each episode (refer to their definitions).

We have verified the exact equivalence between ATD (6)-(7) and every-visit MC updating on random walk experiment (Sutton & Barto, 1998). An interesting observation is that the adaptive stepsize (7) could be very large (even bigger than 2) in the first several episodes. This is possible since the first factor in (7) could be larger than 1, although the second factor is always no larger than 1. As more and more episodes are finished, the adaptive stepsize in (7) will be dominated by the second factor, which is very similar to the stepsize used in (Singh & Sutton, 1996) when proving the equivalence of offline TD(1) and every-visit MC.

3. Conclusion and Future Work

We showed that by extending the adaptive stepsize TD algorithm (Hutter & Legg, 2008) to absorbing MDPs, we actually found an online (every-visit) Monte Carlo algorithm for RL prediction problems. Our result differs from previous online TD algorithm (with any predetermined stepsize) in its *exact* equivalence to the offline every-visit Monte Carlo updating.

Our current result only works for *every-visit* MC updating and we are now trying to extend it to *first-visit* MC by using replacing eligibility trace (Singh & Sutton, 1996) and revising the visiting times $N_s^{(t)}$. Another interesting work is to see how linear function approximation could be applied to the adaptive stepsize TD algorithm.

References

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

Hutter, M., & Legg, S. (2008). Temporal difference updating without a learning rate. In J. Platt, D. Koller, Y. Singer and S. Roweis (Eds.), *Advances in neural information processing systems 20*, 705–712. Cambridge, MA: MIT Press.

Singh, S. P., & Sutton, R. S. (1996). Reinforcement learning with replacing eligibility traces. *Machine Learning*, 123–158.

Sutton, R., & Barto, A. (1998). *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press.