Assessment of Sample-Efficiency of LSTD

[Assignment-4]

(CS 394R: Reinforcement Learning)

Shobhit Chaurasia
Department of Computer Science

December 2, 2016

1 Problem Statement

As established in Section 9.4 of our textbook [1], TD(0) with linear function approximation converges to a fixpoint, for appropriately decreasing step-size, which can be computed can be directly estimated without iterative computation that is performed in TD(0). Least-Squares TD (LSTD) leverages this idea to perform value estimation, and is claimed to be more sample-efficient than TD(0). The motivation behind this assignment is to assess the sample-efficiency of LSTD and compare it with TD(0).

2 Introduction

TD(0) algorithm for value estimation with linear function approximation converges asymptotically to the TD fixpoint given by:

$$\theta_{TD} = A^{-1}b \tag{1}$$

where

$$A = \mathbb{E}\left[\phi_t(\phi_t - \gamma\phi_{t+1})^\top\right], \qquad b = \mathbb{E}\left[R_{t+1}\phi_t\right]$$

The value function is approximated with a function that is linear in the state features $\phi \in \mathbb{R}^n$:

$$\hat{v}(s, \theta) = \theta^\top \phi \tag{2}$$

TD(0) computes this solution iteratively. Alternatively, the fixpoint can be directly estimated by estimating A and b. LSTD uses this idea by using the following natural estimates of A and b:

$$\widehat{A_t} = \sum_{k=0}^{t} \phi_k(\phi_k - \gamma\phi_{k+1})^\top + \epsilon I, \qquad \widehat{b_t} = \sum_{k=0}^{t} R_{t+1}\phi_t$$

where I is an identity matrix, and ϵI ensures that $\widehat{A_t}$ is always invertible. The estimated TD fixpoint is, then, given by:

$$\theta_{t+1} = \widehat{A_t}^{-1}\widehat{b_t} \tag{3}$$

LSTD is claimed to be more sample-efficient than TD(0) because it directly estimates the TD-fixpoint. To validate this claim, the two algorithms are tested in a simple domain with linear function approximators used for value estimation.

3 Experiment

3.1 Setup

The environment, called ChainWorld, consists of a chain of states. There are 1000 states, numbered 1 to 1000 from left to right. All episodes begin near the center: state 500. There are two terminal states: one on the far left (state 0) and one on the far right (state 1000). State transitions can occur from the current state to one of the 100 adjacent states to the right, or to one of the 100 adjacent states to the left, all with equal probability. If there are fewer than 100 states on either side, then the remaining probability mass is assigned to the transition to the terminal state on that side.

Every transition results in zero reward, except transitions to the two terminal states. Transition to the left terminal state (state 0) results in a reward of -1, while transition to the right terminal state results in a reward of 1.

3.2 State Aggregation

State aggregation, a simple form of generalizing function approximation, is used in this domain. States are divided into 10 groups, each containing 100 states, and a single feature — and hence a common value estimate — is used for each group. State aggregation is used to reduce the size of state-space, something that could be prohibitive for LSTD which involves matrix inversion – an operation which, without optimization, has $O(n^3)$ complexity; with optimization, the complexity is still $O(n^2)$.

3.3 Experiment-1

The two algorithms are run for 1000 iterations in the ChainWorld domain. The true values for all states are computed by solving 1000 linear equations. In the experiments reported below, a single invocation of the TD(0) algorithm is compared against multiple invocations of LSTD.

TD(0) with a learning rate of 0.07, decaying by a factor of 0.99 after each episode, is used; the values are chosen through gridsearch. In addition to the sample-efficiency, the lack of learning rate in LSTD is touted as another advantage over TD(0). However, LSTD does have a hyperparameter ϵ. Different invocations of LSTD in the experiments are obtained by varying ϵ.

Fig. 1 shows the value estimates computed by the algorithms against the true value estimates. No single algorithm has an upper hand against the others in terms of the closeness of its value estimates across all state-groups.

To facilitate comparison, Fig. 2 shows the Mean-Squared Value Error (MSVE) as a function of number of episodes. The asymptotic errors of TD(0) and LSTD with $\epsilon = 90$ (LSTD-90) are the lowest, although the differences between the aysmptotic errors of all algorithms are quite small. As evident from the figure, the choice of ϵ has a significant impact on the performance of LSTD, akin to the impact of learning rate on the performance of TD(0). High values of ϵ lead to stable updates, but slow down learning. This is precisely why the error for LSTD with $\epsilon = 90$ smoothly and steadily, albeit slowly, goes down, and converges to the least value among all algorithms. However, it is significantly *less* sample-efficient than TD(0) which attains the asymptotic error by 400 episodes as compared to almost 850 episodes of LSTD-90. As ϵ is lowered, the updates become increasingly less stable. Errors for all LSTD invocations except that for $\epsilon = 90$ drop below that of TD(0) in the first few tens of episodes — hinting at a potential for sample-efficiency in LSTD — but the oscillations due to unstable updates degrade their performance. Of particular interest are LSTD-9 and LSTD-0.009 which achieve lower errors than TD(0) for extended period of few hundred episodes before asymptoting to slightly higher errors.

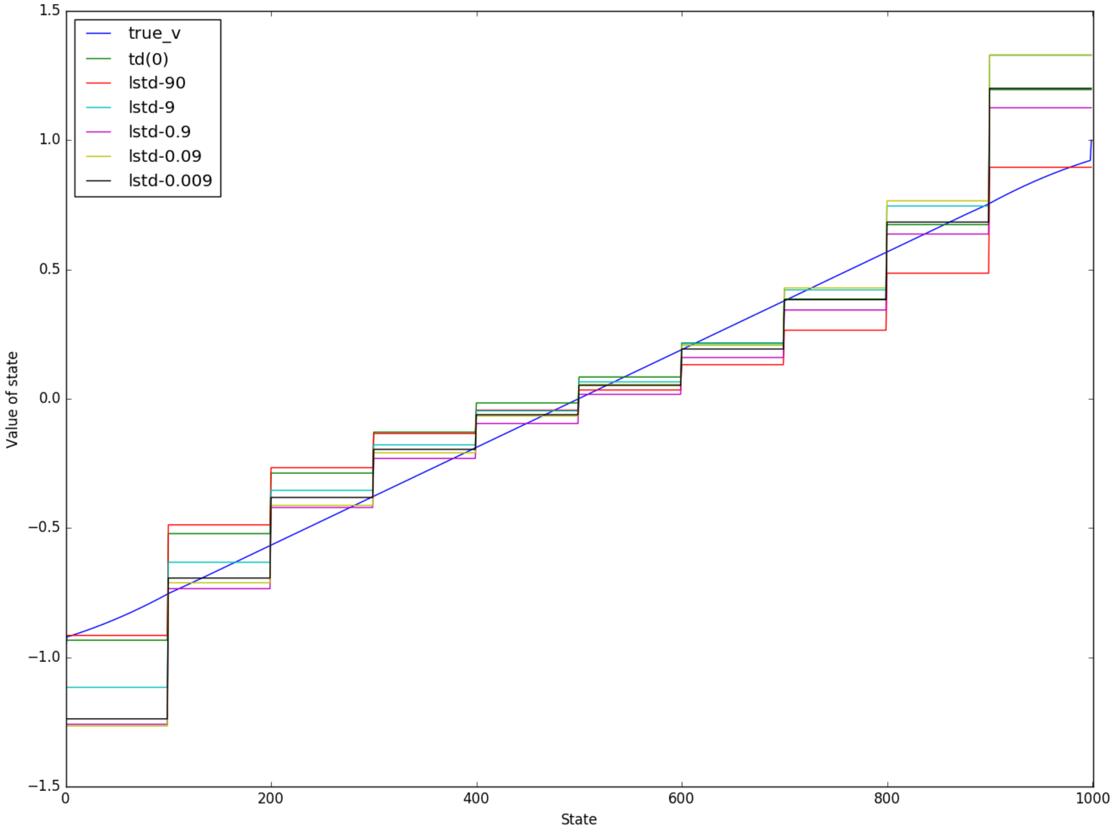

Figure 1: Value estimates for all states computed by different algorithms against the true state-values (denoted by the dark-blue line labeled *true_v*). LSTD-k represents an invocation of LSTD with $\epsilon = k$.

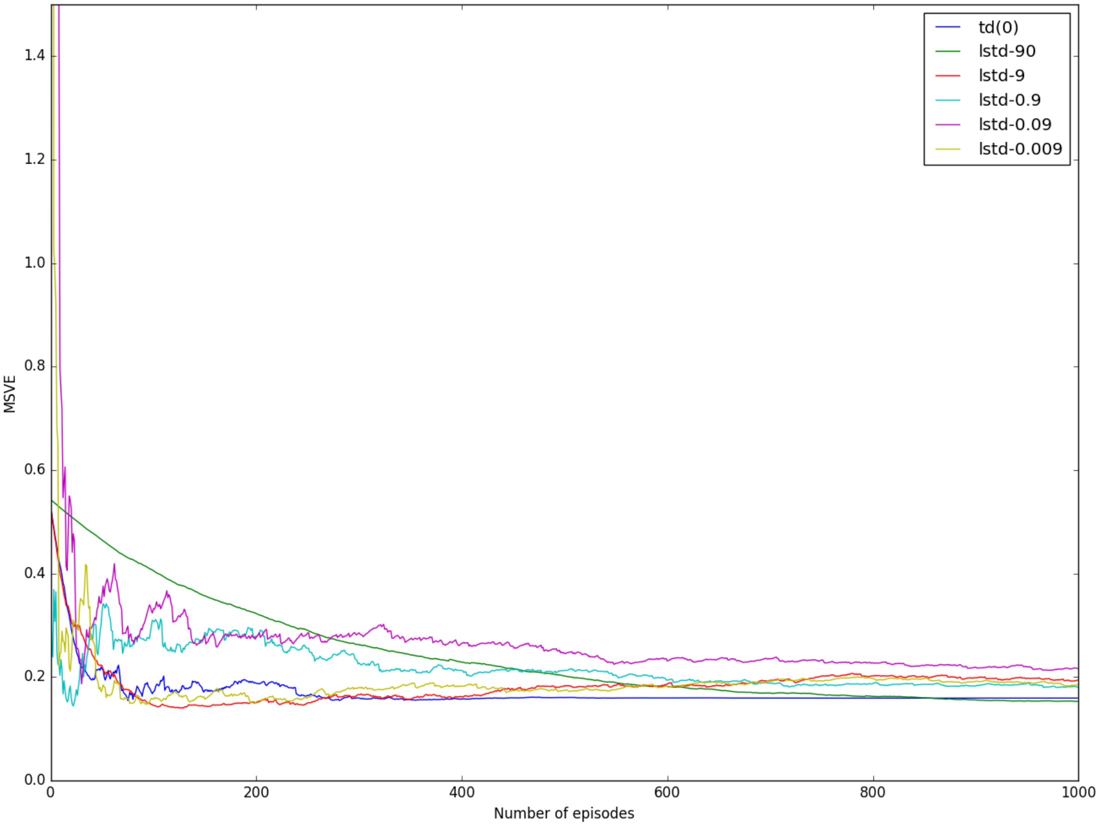

Figure 2: MSVE errors for different algorithms as a function of number of episodes. LSTD-k represents an invocation of LSTD with $\epsilon = k$.

3.4 Experiment - 2

To test the effect of the state-aggregation scheme on the performance of TD(0) and LSTD, in this experiment, the states are grouped into 20 groups of 50 states each. The same algorithms are run with the modified feature vectors for states.

Fig. 3 shows the value estimates computed by the algorithms against the true value estimates. Fig. 4 shows the Mean-Squared Value Error (MSVE) as a function of number of episodes. The absolute errors in value estimates are higher as compared to the original state-aggregation scheme. A clear explanation for this effect seems difficult since there are two competing factors here: lower resolution (fewer state-groups) eases the job of value-estimation because there are fewer ”states,” thereby allowing for more precise value-estimation; however, a lower resolution also means higher number of states grouped together and forced to share a common value, thereby increasing errors in value-estimates.

The steady updates of LSTD-90 are dramatically slow, and it is far from convergence even after 1000 episodes. Others have virtually equal asymptotic errors. The instability associated with lower ϵ values, while present, is less profound. LSTD-0.9, LSTD-0.09, and LSTD-0.009 achieve an error lower than that of TD(0) in fewer episodes, thereby supporting the claim that LSTD is more sample-efficient.

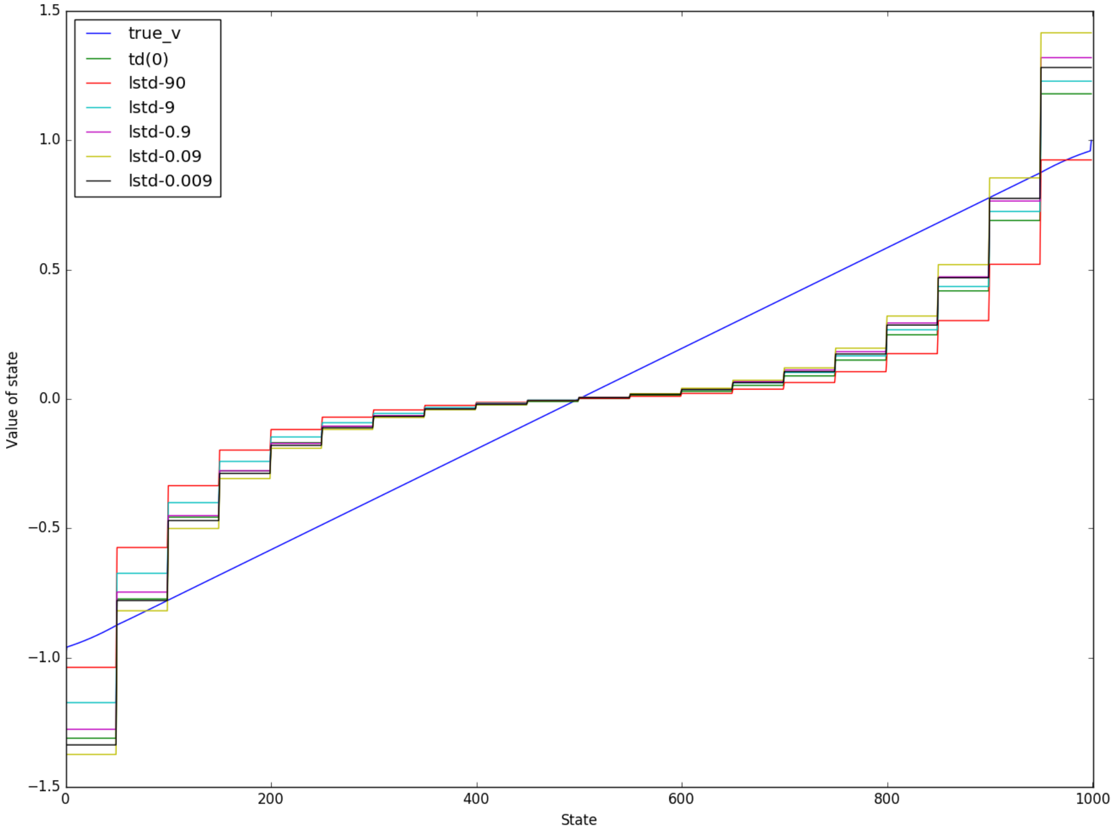

Figure 3: Value estimates for all states computed by different algorithms against the true state-values. Here, state-aggregation scheme of 50-states-per-group is used.

4 Conclusion

The experiments presented here do not provide conclusive evidence either in favor or against the common wisdom that LSTD is more sample-efficient than TD(0). However, one thing that is clear is that ϵ plays as crucial a role in LSTD as the learning rate in TD(0). Hence, it would be premature to give LSTD an edge over TD(0) merely on account of the absence of learning rate in LSTD.

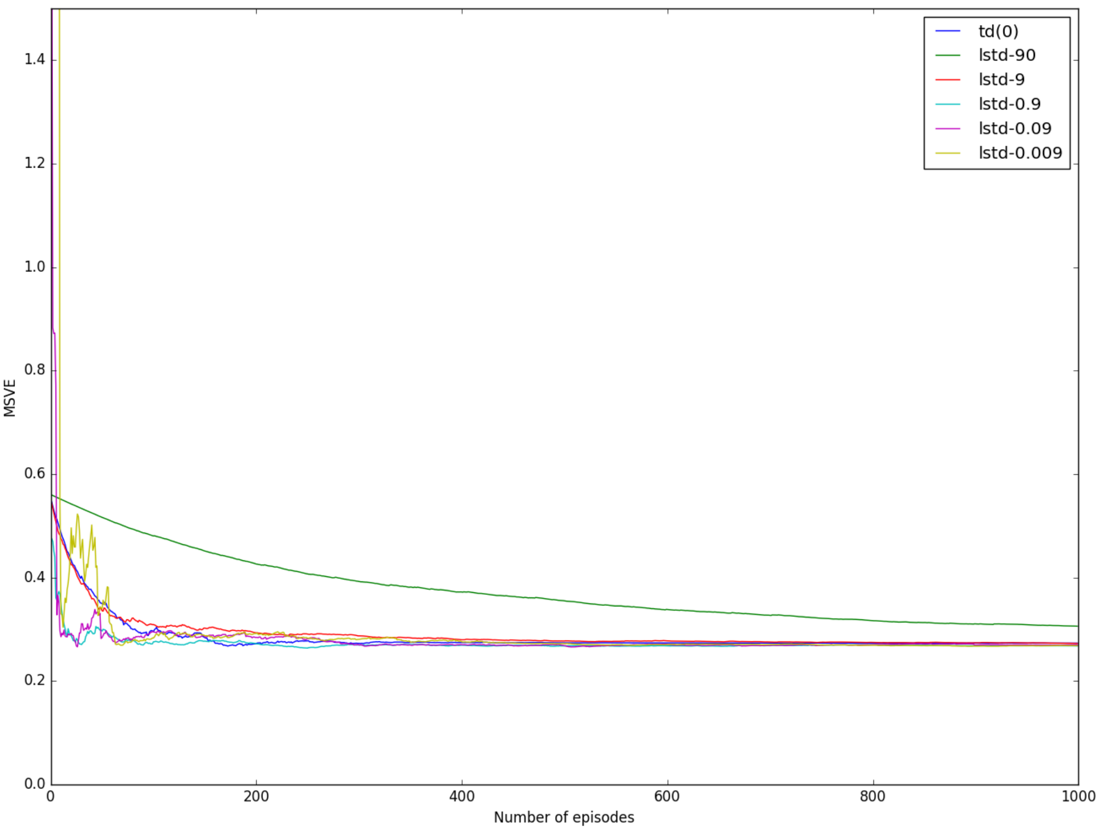

Figure 4: MSVE errors for different algorithms as a function of number of episodes. Here, state-aggregation scheme of 50-states-per-group is used.

References

[1] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. Vol. 1. 1. MIT press Cambridge, 1998.