Bias-Variance Error Bounds for Temporal Difference Updates

Michael Kearns
AT&T Labs
180 Park Avenue, Room A235
Florham Park, NJ 07932
mkearns@research.att.com

Satinder Singh
AT&T Labs
180 Park Avenue, Room A269
Florham Park, NJ 07932
baveja@research.att.com

Abstract

Temporal difference (TD) algorithms are used in reinforcement learning to compute estimates of the value of a given policy in an unknown Markov decision process (policy evaluation). We give rigorous upper bounds on the error of the closely related *phased* TD algorithms (which differ from the standard updates in their treatment of the learning rate) as a function of the amount of experience. These upper bounds prove exponentially fast convergence, with both the rate of convergence and the asymptote strongly dependent on the length of the backups k or the parameter λ. Our bounds give formal verification to the well-known intuition that TD methods are subject to a bias-variance trade-off, and they lead to schedules for k and λ that are predicted to be better than any fixed values for these parameters. We give preliminary experimental confirmation of our theory for a version of the random walk problem.

1 Introduction

In the *policy evaluation* problem, we must predict the expected discounted return (or *value*) for a fixed policy π, given only the ability to generate experience in an unknown Markov decision process (MDP) M. A well-studied parameterized family of *temporal difference* (or TD) [3] algorithms have been developed for this problem. These algorithms make use of repeated trajectories under π from the state(s) of interest, and perform iterative updates to the value function. The parameters of the algorithms control how far they look ahead in the trajectories. The TD(k) algorithm uses the first k rewards, and the (current) value prediction at the $(k+1)$st state reached, in making its update. The more commonly used TD(λ) family of algorithms use exponentially weighted sums of TD(k) updates (with decay parameter λ). The smaller the value for k or λ, the less the algorithm depends on the actual rewards received in the trajectory, and the more it depends on the current predictions for the value function. Conversely, the larger the value for k or λ, the more the algorithm depends on the actual rewards obtained, with the current value function playing a lessened role. The extreme cases of TD$(k=\infty)$ and TD$(\lambda=1)$ become the Monte Carlo algorithm, which updates each prediction to be the average of the discounted returns in the trajectories.

A well-known issue is whether it is better to use large or small values of the parameters k and λ. Watkins [5] informally discusses the trade-off that this decision gives rise to: larger values for the TD parameters suffer larger variance in the updates (since more stochastic reward terms appear), but also enjoy lower bias (since the error in the current value function predictions have less influence). This argument has largely remained an intuition. However, some conclusions arising from this intuition – for instance, that intermediate values of k and λ often yield the best performance in the short term – have been borne out experimentally [4, 2].

In this paper, we provide rigorous upper bounds on the error in the value functions of *phased* TD algorithms as a function of the number of trajectories used. In other words, we give bounds on the *learning curves* of phased TD methods that hold for any MDP. The phased TD algorithms capture the spirit of the standard TD methods, but treat the learning rate in a way that permits a simplified analysis. Our upper bounds decay exponentially fast, and are obtained by first deriving a one-step recurrence relating the errors before and after a phased TD update, and then iterating this recurrence for the desired number of steps. Of particular interest is the form of our bounds, since it formalizes the trade-off discussed above — the bounds consist of terms that are monotonically growing with k and λ (corresponding to the increased variance), and terms that are monotonically shrinking with k and λ (corresponding to the decreased influence of the current error).

Overall, our bounds provide the following contributions and predictions:

- A formal theoretical explanation of the bias-variance trade-off in phased multi-step TD updates;
- A proof of exponentially fast rates of convergence for any fixed k or λ;
- A rigorous upper bound that predicts that larger val-

ues of k and λ lead to faster *convergence*, but to *higher* asymptotic errror;

- Formal explanation of the superiority of intermediate values of k and λ (U-shaped curves) for any fixed number of iterations;
- Derivation of a decreasing *schedule* of k and λ that our bound predicts should beat any fixed value of these parameters.

Furthermore, we provide some preliminary experimental confirmation of our theory for the random walk problem. We note that some of the findings above were conjectured by Singh and Dayan [2] through analysis of specific MDPs.

2 Technical Preliminaries

Let $M = (P, R)$ be an MDP, consisting of the *transition distributions* $P(\cdot|s,a)$ over next states for any state-action pair (s,a), and the *reward distributions* $R(\cdot|s)$ over scalar rewards at each state s. For any policy π (a mapping from states to actions) in M, and any start state s_0, a *trajectory* generated by π starting from s_0 is a random variable τ that is an infinite sequence of states and rewards:

$$\tau = (s_0, r_0) \to (s_1, r_1) \to (s_2, r_2) \to \cdots .$$

Here each random reward r_i is distributed according to $R(\cdot|s_i)$, and each state s_{i+1} is distributed according to $P(\cdot|s_i, \pi(s_i))$. For simplicity we will assume that the support of $R(\cdot|s_i)$ is $[-1,+1]$. However, all of our results easily generalize to the case of bounded variance. We define the *value function* of π, $V^\pi(s)$, to be the expected discounted return of trajectories τ generated by π starting from s:

$$V^\pi(s) = \mathbf{E}_\tau[r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots]$$

where $0 \leq \gamma < 1$ is a fixed *discount factor*. One of the central problems in reinforcement learning is that of estimating the value function of a fixed π on the basis of sample trajectories under π.

We now define the standard $\mathrm{TD}(k)$ (also known as *k-step backup*) and $\mathrm{TD}(\lambda)$ algorithms for updating an estimate of the value function. Given a trajectory τ generated by π from s, and given an estimate $\hat{V}^\pi(\cdot)$ for the value function $V^\pi(\cdot)$, for any natural number k we define

$$\begin{aligned} \mathrm{TD}(k, \tau, \hat{V}^\pi(\cdot)) \quad &= \quad (1-\alpha)\hat{V}^\pi(s) \\ &+\alpha\left(r_0 + \gamma r_1 + \cdots + \gamma^{k-1} r_{k-1} + \gamma^k \hat{V}^\pi(s_k)\right) . \end{aligned}$$

The $\mathrm{TD}(k)$ update based on τ is simply

$$\hat{V}^\pi(s) \leftarrow \mathrm{TD}(k, \tau, \hat{V}^\pi(\cdot)).$$

It is implicit that the update is always applied to the estimate at the initial state of the trajectory τ, and we regard the discount factor γ and the *learning rate* α as being fixed. For any $\lambda \in [0,1]$, the $\mathrm{TD}(\lambda)$ update can now be easily expressed as an infinite linear combination of the $\mathrm{TD}(k)$ updates:

$$\mathrm{TD}(\lambda, \tau, \hat{V}^\pi(\cdot)) = \sum_{k=1}^{\infty} (1-\lambda)\lambda^{k-1} \mathrm{TD}(k, \tau, \hat{V}^\pi(\cdot)).$$

Given a sequence $\tau_1, \tau_2, \tau_3, \ldots$, we can simply apply either type of TD update sequentially. In either case, as either k becomes large or λ approaches 1, the updates approach a Monte Carlo method, in which we use each trajectory τ_i entirely, and ignore our current estimate $\hat{V}^\pi(\cdot)$. As k becomes small or λ approaches 0, we rely heavily on the estimate $\hat{V}^\pi(\cdot)$, and effectively use only a few steps of each τ_i. The common intuition is that early in the sequence of updates, the estimate $\hat{V}^\pi(\cdot)$ is poor, and we are better off choosing k large or λ near 1. However, since the trajectories τ_i do obey the statistics of π, the value function estimates will eventually improve, at which point we may be better off "bootstrapping" by choosing small k or λ.

In order to provide a rigorous analysis of this intuition, we will study what we call *phased* TD updates. These algorithms are intended to capture the qualitative properties of the standard TD methods, while simplifying the complexities of the moving average introduced by the learning rate α. In each phase, we are given n trajectories under π from every state s, where n is a parameter of the analysis. Thus, phase t consists of a set $S(t) = \{\tau_i^s(t)\}_{s,i}$, where s ranges over all states, i ranges from 1 to n, and $\tau_i^s(t)$ is an independent random trajectory generated by π starting from state s. In phase t, phased TD averages all n of the trajectories in $S(t)$ that start from state s to obtain its update of the value function estimate for s. In other words, the phased $\mathrm{TD}(k)$ updates become

$$\begin{aligned} \hat{V}_{t+1}^\pi(s) \quad &\leftarrow \quad (1/n)\sum_{i=1}^{n} \Big(r_0^i + \gamma r_1^i + \cdots \\ &\qquad + \gamma^{k-1} r_{k-1}^i + \gamma^k \hat{V}_t^\pi(s_k^i)\Big) \end{aligned}$$

where the r_j^i are the rewards along trajectory $\tau_i^s(t)$, and s_k^i is the kth state reached along that trajectory. The phased $\mathrm{TD}(\lambda)$ updates become

$$\begin{aligned} \hat{V}_{t+1}^\pi(s) \quad &\leftarrow \quad (1/n)\sum_{i=1}^{n} \left(\sum_{k=1}^{\infty} (1-\lambda)\lambda^{k-1} \Big(r_0^i + \gamma r_1^i + \cdots \right. \\ &\qquad \left. + \gamma^{k-1} r_{k-1}^i + \gamma^k \hat{V}_t^\pi(s_k^i)\Big)\right) \end{aligned}$$

Note that the phased TD updates are subject to the same bias-variance intuition as the standard updates. Indeed, we view phased TD updates with a constant value of n as roughly analogous to standard TD updates with a constant learning rate α [1], with larger n corresponding to smaller α. To see this, note that we may "unroll" the standard $\mathrm{TD}(k)$ estimate after t iterations as

$$\begin{aligned} \hat{V}_t^\pi(s) \quad &= \quad (1-\alpha)^t \hat{V}_0^\pi(s) + (1-\alpha)^{t-1}\alpha(\rho_0 + \mu_0) \\ &+ (1-\alpha)^{t-2}\alpha(\rho_1 + \mu_1) + \cdots + \alpha(\rho_{t-1} + \mu_{t-1}). \end{aligned}$$

Here we use $\rho_i = (r_0^i + \gamma r_1^i + \cdots + \gamma^{k-1} r_{k-1}^i)$ and $\mu_i = \gamma^k \hat{V}_t^\pi(s_k^i)$. Thus, for any *fixed* α, because of the exponential damping, our estimate at any given moment is directly dependent on a number of recent trajectories that is effectively constant. We note that since it is common in practice to use a decreasing (and not constant) learning rate, we are analyzing here an algorithm that in at least one way is believed to be inferior to those used experimentally.

In the ensuing sections, we provide a rigorous upper bound on the error in the value function estimates of phased TD updates as a function of the number of phases. This upper bound clearly captures the bias-variance intuitions expressed above. We note that while the experimental and theoretical relationship between standard and phased TD updates needs to be explored, the phased TD algorithms are well-defined, simple and easily implemented in their own right, and to the extent that one believes the standard updates to be *superior* to the phased updates, our upper bounds are relevant to the former.

3 Bounding the Error of Phased TD Updates

Theorem 1 *(Phased* $\mathrm{TD}(k)$ *Error Recurrence) Let* $S(t)$ *be the set of trajectories generated by* π *in phase* t *(*n *trajectories from each state), let* $\hat{V}_t^\pi(\cdot)$ *be the value function estimate of phased* $\mathrm{TD}(k)$ *after phase* t*, and let*

$$\Delta_t = \max_s \{ |\hat{V}_t^\pi(s) - V^\pi(s)| \}.$$

Then for any $1 > \delta > 0$*, with probability at least* $1 - \delta$*,*

$$\Delta_t \leq \frac{1-\gamma^k}{1-\gamma} \sqrt{\frac{3\log(k/\delta)}{n}} + \gamma^k \Delta_{t-1}. \qquad (1)$$

Here the error Δ_{t-1} *after phase* $t-1$ *is fixed, and the probability is taken over only the trajectories in* $S(t)$*.*

Proof:(Sketch) We begin by writing

$$\begin{aligned} V^\pi(s) &= \mathbf{E}[r_0 + \gamma r_1 + \cdots + \gamma^{k-1} r_{k-1} + \gamma^k V^\pi(s_k)] \\ &= \mathbf{E}[r_0] + \gamma \mathbf{E}[r_1] + \cdots + \gamma^{k-1}\mathbf{E}[r_{k-1}] \\ &\qquad + \gamma^k \mathbf{E}[V^\pi(s_k)]. \end{aligned}$$

Here the expectations are over a random trajectory under π; thus $\mathbf{E}[r_\ell]$ ($\ell \leq k-1$) denotes the expected value of the ℓth reward received, while $\mathbf{E}[V^\pi(s_k)]$ is the expected value of the true value function at the kth state reached. The phased $\mathrm{TD}(k)$ update sums the terms $\gamma^\ell (1/n) \sum_{i=1}^n r_\ell^i$, whose expectations are exactly the $\gamma^\ell \mathbf{E}[r_\ell]$ appearing above. By a standard large deviation analysis (omitted), the probability that any of these terms deviate by more than

$$\epsilon = \sqrt{3\log(k/\delta)/n}$$

from their expected values is at most δ. If no such deviation occurs, the total contribution to the error in the value function estimate is bounded by $((1-\gamma^k)/(1-\gamma))\epsilon$, giving rise to the "variance" term in our overall bound above. The remainder of the phased $\mathrm{TD}(k)$ update is simply $\gamma^k (1/n) \sum_{i=1}^n \hat{V}_{t-1}^\pi(s_k^i)$. But since $|\hat{V}_{t-1}^\pi(s_k^i) - V^\pi(s_k^i)| \leq \Delta_{t-1}$ by definition, the contribution to the error is at most $\gamma^k \Delta_{t-1}$, which is the "bias" term of the bound. We note that a similar argument leads to bounds in expectation rather than the PAC-style bounds given here. □

Let us take a brief moment to analyze the qualitative behavior of Equation (1) as a function of k. For large values of k, the quantity γ^k becomes negligible, and the bound is approximately $(1/(1-\gamma))\sqrt{3\log(k/\delta)/n}$, giving almost all the weight to the error incurred by variance in the first k rewards, and negligible weight to the error in our current value function. At the other extreme, when $k = 1$ our reward variance contributes error only $\sqrt{3\log(1/\delta)/n}$, but the error in our current value function has weight γ. Thus, the first term increases with k, while the second term decreases with k, in a manner that formalizes the intuitive trade-off that one faces when choosing between longer or shorter backups.

Equation (1) describes the effect of a single phase of $\mathrm{TD}(k)$ backups, but we can iterate this recurrence over many phases to derive an upper bound on the full learning curve for any value of k. Assuming that the recurrence holds for t consecutive steps, [1] and assuming $\Delta_0 = 1$ without loss of generality, solution of the recurrence (details omitted) yields

$$\Delta_t \leq \frac{1-\gamma^{kt}}{1-\gamma}\sqrt{3\log(k/\delta)/n} + \gamma^{kt}. \qquad (2)$$

This bound makes a number of predictions about the effects of different values for k. First of all, as t approaches infinity, the bound on Δ_t approaches the value

$$(1/(1-\gamma))\sqrt{3\log(k/\delta)/n},$$

which increases with k. Thus, the bound predicts that *the asymptotic error of phased* $\mathrm{TD}(k)$ *updates is larger for larger* k [2]. On the other hand, the *rate* of convergence to this asymptote is γ^{kt}, which is always exponentially fast, but *faster* for larger k. Thus, in choosing a fixed value of k, we must choose between having either rapid convergence to a worse asymptote, or slower convergence to a better asymptote. This prediction is illustrated graphically in Figure 1(a), where with all of the parameters besides k and t fixed (namely, γ, δ, and n), we have plotted the bound of Equation (2) as a function of t for several different choices of k.

Note that while the plots of Figure 1(a) were obtained by choosing *fixed* values for k and iterating the recurrence of Equation (1), at each phase t we can instead use Equation (1) to choose the value of k that maximizes the predicted

[1] Formally, we can apply Theorem 1 by choosing $\delta = \delta'/(tN)$, where N is the number of states in the MDP. Then with probability at least $1-\delta'$, the bound of Equation (1) will hold at every state for t consecutive steps.

[2] We note that this statement is valid for the case of constant n, which is analogous to a constant learning rate, as already discussed. Decreasing learning rates can of course achieve zero asymptotic error.

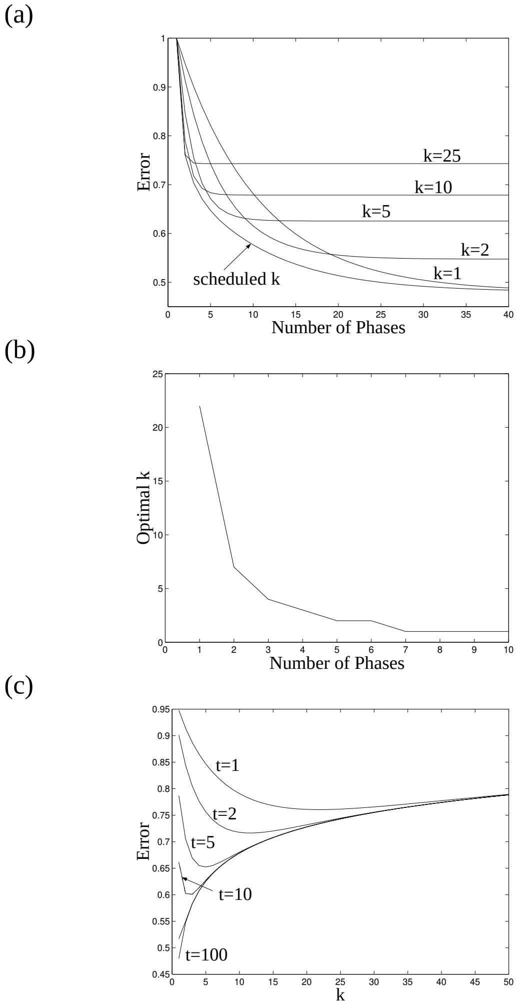

Figure 1: (a) Upper bounds on the learning curves Δ_t of phased TD(k) for several values of k, as a function of the number of phases t (parameters $n = 3000, \gamma = 0.9, \delta = 0.1$). Note that larger values of k lead to more rapid convergence, but to higher asymptotic errors. Both the theory and the curves suggest a (decreasing) schedule for k, intuitively obtained by always "jumping" to the learning curve that enjoys the greatest one-step decrease from the current error. This schedule can be efficiently computed from the analytical upper bounds, and leads to the best (lowest) of the learning curves plotted, which is significantly better than for any fixed k. (b) The schedule for k derived from the theory as a function of the number of phases t. (c) For several values of the number of phases t, the upper bound on Δ_t for phased TD(k) as a function of k. These curves show the predicted trade-off, with a unique optimal value for k identified until t is sufficiently large to permit 1-step backups to converge to their optimal asymptotes.

decrease in error $\Delta_t - \Delta_{t+1}$. In other words, the recurrence immediately yields a *schedule* for k, along with an upper bound on the learning curve for this schedule that outperforms the upper bound on the learning curve for any fixed value of k. The learning curve for the schedule is also shown in Figure 1(a), and Figure 1(b) plots the schedule itself.

Another interesting set of plots is obtained by fixing the number of phases t, and computing for each k the error after t phases using TD(k) updates that is predicted by Equation (2). Such plots are given in Figure 1(c), and they clearly predict a unique minimum — that is, an optimal value of k for each fixed t (this can also be verified analytically from equation 2). For moderate values of t, values of k that are too small suffer from their overemphasis on a still-inaccurate value function approximation, while values of k that are too large suffer from their refusal to bootstrap. Of course, as t increases, the optimal value of k decreases, since small values of k have time to reach their superior asymptotes.

We now go on to provide a similar analysis for the TD(λ) family of updates, beginning with the analogue to Theorem 1.

Theorem 2 *(Phased* TD(λ) *Error Recurrence) Let $S(t)$ be the set of trajectories generated by π in phase t (n trajectories from each state), let $\hat{V}_t^\pi(\cdot)$ be the value function estimate of phased* TD(λ) *after phase t, and let*

$$\Delta_t = \max_s\{|\hat{V}_t^\pi(s) - V^\pi(s)|\}.$$

Then for any $1 > \delta > 0$, with probability at least $1 - \delta$,

$$\Delta_t \leq \min_k \left\{ \frac{1-(\gamma\lambda)^k}{1-\gamma\lambda}\sqrt{\frac{3\log(k/\delta)}{n}} + \frac{(\gamma\lambda)^k}{1-\gamma\lambda} \right\} + \frac{(1-\lambda)\gamma}{1-\gamma\lambda}\Delta_{t-1}. \qquad (3)$$

Here the error Δ_{t-1} after phase $t-1$ is fixed, and the probability is taken over only the trajectories in $S(t)$.

We omit the proof of this theorem, but it roughly follows that of Theorem 1. That proof exploited the fact that in TD(k) updates, we only need to apply large deviation bounds to the rewards of a finite number (k) of averaged trajectory steps. In TD(λ), *all* of the rewards contribute to the update. However, we can always choose to bound the deviations of the first k steps, for any value of k, and assume maximum variance for the remainder (whose weight diminishes rapidly as we increase k). This logic is the source of the $\min_k\{\cdot\}$ term of the bound. One can view Equation (3) as a *variational* upper bound, in the sense that it provides a family of upper bounds, one for each k, and then minimizes over the variational parameter k.

The reader can verify that the terms appearing in Equation (3) exhibit a trade-off as a function of λ analogous to that exhibited by Equation (1) as a function of k. In the interest of brevity, we move directly to the TD(λ) analogue of Equation (2). It will be notationally convenient to define

$k_\lambda = \mathrm{argmin}_k\{F(\lambda)\}$, where $F(\lambda)$ is the function appearing inside the $\min_k\{\cdot\}$ in Equation (3). (Here we regard all parameters other than λ as fixed.) It can be shown that for $\Delta_0 = 1$, repeated iteration of Equation (3) yields the t-phase inequality

$$\Delta_t \leq a_\lambda \frac{1 - {b_\lambda}^t}{1 - b_\lambda} + {b_\lambda}^t \tag{4}$$

where

$$a_\lambda = \frac{1-(\gamma\lambda)^{k_\lambda}}{1-\gamma\lambda}\sqrt{\frac{3\log(k_\lambda/\delta)}{n}} + \frac{(\gamma\lambda)^{k_\lambda}}{1-\gamma\lambda}$$

and

$$b_\lambda = \frac{(1-\lambda)\gamma}{1-\gamma\lambda}.$$

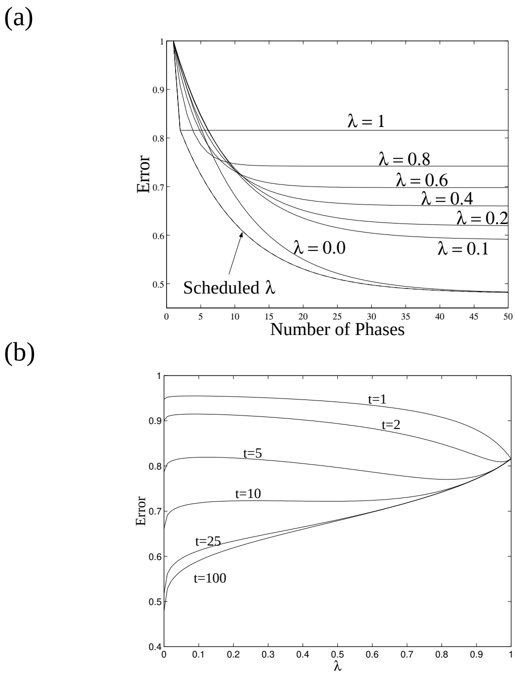

Figure 2: (a) Upper bounds on the learning curves Δ_t of phased $\mathrm{TD}(\lambda)$ for several values of λ, as a function of the number of phases t (parameters $n = 3000$,$\gamma = 0.9, \delta = 0.1$). The predictions are analogous to those for $\mathrm{TD}(k)$ in Figure 1, and we have again plotted the predicted best learning curve obtained via a decreasing schedule of λ. (b) For several values of the number of phases t, the upper bound on Δ_t for $\mathrm{TD}(\lambda)$ as a function of λ.

While Equation (4) may be more difficult to parse than its $\mathrm{TD}(k)$ counterpart, the basic predictions and intuitions remain intact. As t approaches infinity, the bound on Δ_t asymptotes at $a_\lambda/(1 - b_\lambda)$, and the rate of approach to this asymptote is simply ${b_\lambda}^t$, which is again exponentially fast. Analysis of the derivative of b_λ with respect to λ confirms that for all $\gamma < 1$, b_λ is a decreasing function of λ — that is, the larger the λ, the faster the convergence. Analytically verifying that the asymptote $a_\lambda/(1 - b_\lambda)$ increases with λ is more difficult due to the presence of k_λ, which involves a minimization operation. However, the learning curve plots of Figure 2(a) clearly show the predicted phenomena — increasing λ yields faster convergence to a worse asymptote. As we did for the $\mathrm{TD}(k)$ case, we use our recurrence to derive a schedule for λ; Figure 2(a) also shows the predicted improvement in the learning curve by using such a schedule. Finally, Figure 2(b) again shows the non-monotonic predicted error as a function of λ for a fixed number of phases.

4 Some Experimental Confirmation

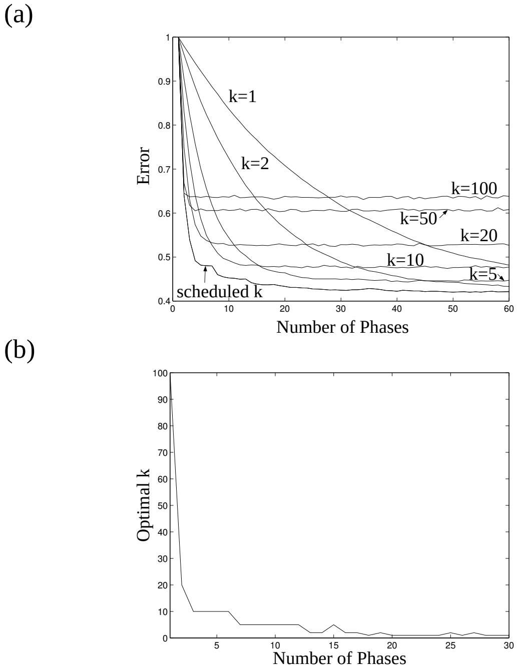

Figure 3: (a) Empirical learning curves Δ_t for $\mathrm{TD}(k)$ for several values of k on the random walk problem (parameters $n = 40$ and $\gamma = 0.98$). Each plot is averaged over 5000 runs of $\mathrm{TD}(k)$. Also shown is the learning curve (averaged over 5000 runs) for the empirical schedule computed from the $\mathrm{TD}(k)$ learning curves, which is better than any of these curves. (b) The empirical schedule.

In order to test the various predictions made by our theory, we have performed a number of experiments using phased $\mathrm{TD}(k)$ on a version of the so-called *random walk* problem [4]. In this problem, we have a Markov process with 5 states arranged in a ring. At each step, there is probability 0.05 that we remain in our current state, and probability 0.95 that we advance one state clockwise around the ring. (Note that since we are only concerned with the evaluation of a fixed policy, we have simply defined a Markov process rather than a Markov decision process.) Two adjacent states on the ring have reward $+1$ and -1 respectively, while the remaining states have reward 0. The standard random walk problem has a chain of states, with an absorbing state at each end; here we chose a ring structure simply to avoid asymmetries in the states induced by the absorbing states.

To test the theory, we ran a series of simulations computing the TD(k) estimate of the value function in this Markov process. For several different values of k, we computed the error Δ_t in the value function estimate as a function of the number of phases t. (Δ_t is easily computed, since we can compute the true value function for this simple problem.) The resulting plot in Figure 3(a) is the experimental analogue of the theoretical predictions in Figure 1(a). We see that these predictions are qualitatively confirmed — larger k leads to faster convergence to an inferior asymptote.

Given these empirical learning curves, we can then compute the "empirical schedule" that they suggest. Namely, to determine experimentally a schedule for k that should outperform (at least) the values of k we tested in Figure 3(a), we used the empirical learning curves to determine, for any given value of Δ, which of the empirical curves enjoyed the greatest one-step decrease in error when its current error was (approximately) Δ. This is simply the empirical counterpart of the schedule computation suggested by the theory described above, and the resulting experimental learning curve for this schedule is also shown in Figure 3(a), and the schedule itself in Figure 3(b). We see that there are significant improvements in the learning curve from using the schedule, and that the form of the schedule is qualitatively similar to the theoretical schedule of Figure 1(b).

5 Conclusion

We have given the first provable upper bounds on the error of TD methods for policy evaluation. These upper bounds have exponential rates of convergence, and clearly articulate the bias-variance trade-off that such methods obey.

References

[1] M. Kearns and S. Singh *Finite-Sample Convergence Rates for Q-Learning and Indirect Algorithms* NIPS, 1998.

[2] S. Singh and P. Dayan *Analytical Mean Squared Error Curves for Temporal Difference Learning.* Machine Learning, 1998.

[3] R. S. Sutton *Learning to Predict by the Methods of Temporal Differences.* Machine Learning, 3, 9-44, 1988.

[4] R. S. Sutton and A. G. Barto *Reinforcement Learning: An Introduction.* MIT Press, 1998.

[5] C.J.C.H. Watkins *Learning from Delayed Rewards.* Cambridge Univ., England, 1989.