Analytical Mean Squared Error Curves for Temporal Difference Learning

SATINDER SINGH baveja@cs.colorado.edu
Department of Computer Science
University of Colorado
Boulder, CO 80309-0430

PETER DAYAN dayan@ai.mit.edu
Department of Brain and Cognitive Sciences
Massachusetts Institute of Technology
Cambridge, MA 02139

Editor: Andrew G. Barto

Abstract. We provide analytical expressions governing changes to the bias and variance of the lookup table estimators provided by various Monte Carlo and temporal difference value estimation algorithms with offline updates over trials in absorbing Markov reward processes. We have used these expressions to develop software that serves as an analysis tool: given a complete description of a Markov reward process, it rapidly yields an exact mean-square-error curve, the curve one would get from averaging together sample mean-square-error curves from an infinite number of learning trials on the given problem. We use our analysis tool to illustrate classes of mean-square-error curve behavior in a variety of example reward processes, and we show that although the various temporal difference algorithms are quite sensitive to the choice of step-size and eligibility-trace parameters, there are values of these parameters that make them similarly competent, and generally good.

Keywords: reinforcement learning, temporal difference, Monte Carlo, MSE, bias, variance, eligibility trace, Markov reward process

1. Introduction

Many different algorithms have been developed for predicting the expected outcome, or value, of uncontrolled Markov reward processes: Monte Carlo (MC) algorithms (e.g., Wasow, 1952) and maximum-likehood (ML) algorithms (e.g., Kumar & Varaiya, 1986) in statistics and control, and temporal difference (TD) algorithms (Sutton, 1988; Barto et al., 1983) in machine learning. For most such algorithms, a theory of asymptotic convergence with probability one is available under suitable conditions on algorithm parameters. However, what is not available is a theory of learning behavior of the kind that is available in some supervised learning problems (e.g., Haussler et al., 1994). For example, which algorithm and problem parameters are key determinants of learning behavior?[1] How do different parameters for the Markov reward process, such as the mixing rate, the amount of determinism, acyclicity, etc., change learning curves? How do these problem parameters interact with algorithm parameters such as the step-size, α, and, in the case of TD, the eligibility-trace parameter, λ? Understanding the effects of these parameters is also crucial to making useful comparisons between algorithms, as it is quite likely that no one algorithm dominates the others for all problems. This understanding will also form a basis for developing hybrid algorithms, and for developing methods that set algorithm parameters automatically for faster learning.

One could address the above questions empirically by studying the learning curves for various algorithms applied to specific, carefully chosen, problems. The difficulty is that the sequence of value estimates produced by both MC and TD algorithms is random, and therefore the learning curves themselves are random. Nevertheless, one could hope to draw sensible conclusions by studying "mean" learning curves produced by averaging a large number of random learning curves. However, one would expect this to be computationally infeasible, except for small problems, and indeed we show below that even for very small problems (e.g., with just 5 states) the distribution of random learning curves may be such as to render the empirical method infeasible. In this paper we provide an analytical way of computing mean learning curves.

We focus on the mean squared error (MSE) between the estimated and true predictions.[2] Our main contribution is in deriving the analytical update equations for the two components of the MSE, the bias and the variance, for popular MC and TD algorithms. Given the mean and covariance matrix of a current guess for the true value and a particular choice of algorithm parameters, our results tell us precisely what the expected MSE is after another trial as a function of the problem parameters. These derivations are based on five assumptions: that the Markov reward process is absorbing, i.e., has terminal states, that lookup tables are used, that the algorithm parameters α and λ are functions of the trial number alone rather than also depending on the state, that the estimated values are updated offline (after the end of each trial), and that the only non-zero payoffs are on the transitions to the terminal states. The effect of violating any of these assumptions on the general nature of our results is not known. With the above caveats, given a complete description of a Markov reward process, our results allow us to rapidly compute *exact* MSE learning curves for MC or TD algorithms as a function of trial number — the same curves one would get by averaging an infinite number of sample MSE learning curves obtained by repeatedly running the learning algorithm on the same Markov reward process.

While our analysis method does not suggest a new learning algorithm, we use it in this paper to produce analytical learning curves for a number of specific Markov reward processes chosen to highlight the effect of various problem and algorithm parameters, in particular different choices of α and λ. Using these learning curves, we also compare the relative performance of different forms of eligibility traces in TD algorithms, as well as the relative performance of TD and MC algorithms. These results are on specific problems, and any conclusions drawn from them are valid only on the problems presented. However, we believe that many of the conclusions are intuitive or have previous empirical support, and may be more generally applicable.

The remainder of the paper is organised as follows. Section 2 describes the problem of estimating the values of states in absorbing, Markov reward processes, and the various MC and TD algorithms we have considered. Section 3 introduces the main results of the paper, namely the update equations for bias and variance of the estimates, which are given in full in the appendix and in the associated software. Section 4 applies the software to certain specific Markov reward processes to determine the effects of the different parameters of the algorithms. Section 5 analyses what these bias and variance update formulæ imply about the asymptotic convergence rates for the algorithms, at least for constant learning rates. Finally, section 6 draws together the conclusions.

2. The Value Prediction Problem and Learning Algorithms

We consider absorbing Markov reward processes with a finite set of non-terminal states $s = 1, \ldots, n$. The probability of a transition from non-terminal state i to non-terminal state j is denoted by Q_{ij} and the probability of absorption from i is denoted by q_i. There is no payoff on transitions between non-terminal states. On absorption from state i there is a random payoff, denoted r_i, whose expected value is a function of i. The prediction problem is to determine the value of every non-terminal state i, denoted v_i^*, defined as the expected terminal payoff when the start state is i. Therefore, $v_i^* = E\{r|s_1 = i\}$, where s_k is the state at step k, and r is the random terminal payoff.

Both TD and MC algorithms begin with an initial guess of the value function and use learning trials to update their guesses. A learning trial consists of a random walk that starts in state i with probability μ_i and produces a sequence of non-terminal states followed by a terminal payoff. The update equations of all of the algorithms analyzed take the following general form, for all i:

$$v_i(t) = v_i(t-1) + \alpha(t)\delta_i(t), \tag{1}$$

where the vector $\mathbf{v}(t) = \{v_i(t)\}$ is the estimate of the value function after t trials, $\delta_i(t)$ is the estimate of the error in $v_i(t-1)$ for state i based on trial t, and the scalar step-size $\alpha(t)$ determines how the error is used to improve the old estimate. The estimate of the error $\delta_i(t)$ might depends on all the values $\mathbf{v}(t-1)$. The algorithms differ in the δs produced from a trial. In general, the initial estimate $\mathbf{v}(0)$ could be a random vector drawn from some distribution, but often $\mathbf{v}(0)$ is fixed to some initial value such as zero. In either case, subsequent estimates, $\mathbf{v}(t), t > 0$, are random vectors because of the random δs.

The bias in the estimate after t trials, $\mathbf{b}(t)$, is defined as $E\{\mathbf{v}(t) - \mathbf{v}^*\}$, i.e., the expected difference between the estimated and the true value. Similarly, the covariance matrix of the estimate after t trials, $C(t)$, is defined as $E\{(\mathbf{v}(t) - E\{\mathbf{v}(t)\})(\mathbf{v}(t) - E\{\mathbf{v}(t)\})^T\}$. If $\mathbf{v}(0)$ is fixed, $\mathbf{b}(0) = \mathbf{v}(0) - \mathbf{v}^*$ and $C(0)$ is the null matrix (with all entries zero). A key scalar quantity of interest is the weighted MSE as a function of trial number t:

$$\mathrm{MSE}(t) \;=\; \sum_i p_i(E\{(v_i(t) - v_i^*)^2\}) = \sum_i p_i(b_i^2(t) + C_{ii}(t)), \tag{2}$$

where the expected squared error for state i is weighted by a scalar p_i. Hereafter, we will only consider weighted MSE and refer to it simply as MSE. We take p_i to be the expected number of visits to i in a trial divided by the expected length of a trial:

$$p_i \overset{\text{def}}{=} \frac{(\mu^T[I-Q]^{-1})_i}{\sum_j (\mu^T[I-Q]^{-1})_j}.$$

Other reasonable choices for the weights, $\{p_i\}$, would not change the nature of the results presented here.

2.1. Learning Algorithms

This section presents all the learning algorithms we study in this paper. Let the indicator variable $K_i(t)$ be one if state i is visited at least once in trial t, and zero otherwise; let $\kappa_i(t)$ be the number of visits to state i in trial t; and let $\tau(t)$ denote the number of time steps in trial t. Note that trial t produces a sequence of $\tau(t)$ states followed by a random terminal payoff $r(t)$.

Monte Carlo (MC)

Monte Carlo algorithms use the terminal payoff that results from a trial to define the δ in Equation 1. Therefore in MC algorithms the estimated value of one state is unaffected by the estimated value of any other state. We study two MC algorithms (Singh & Sutton, 1996):

first-visit MC:

$$v_i(t) = v_i(t-1) + \alpha(t) K_i(t) \left(r(t) - v_i(t-1) \right), \text{ and} \tag{3}$$

every-visit MC:

$$v_i(t) = v_i(t-1) + \alpha(t) \kappa_i(t) \left(r(t) - v_i(t-1) \right). \tag{4}$$

In the case of Markov reward processes with only terminal payoffs, as above, the only difference between first-visit MC and every-visit MC is in the random rescaling of the step-sizes[3] in every-visit MC.

Temporal Difference (TD)

The main difference between TD algorithms (Sutton, 1988) and MC algorithms is that the former update the value of a state based not only on the terminal payoff but also on the the estimated values of the intervening states. When a state is first visited it initiates a short-term memory process, a state-specific eligibility trace, which then decays exponentially over time with parameter λ. The manner in which the values of intervening states are combined with the terminal payoff is determined in part by the magnitudes of the eligibility traces. We study three TD algorithms differing only in the method by which the eligibility trace for a state is updated on *re*visits to the state before termination. As shown in Figure 1, *accumulate* TD adds a new trace to the existing trace, *replace* TD replaces the old trace by a new trace, while *first* TD's trace ignores revisits. Accumulate TD is the original TD algorithm defined by Sutton (1988), replace TD was defined by Singh & Sutton (1996), and we introduce first TD here.

The estimated error for state i after trial t, $\delta_i(t)$ in Equation 1, takes the following form for all three TD algorithms:

$$\delta_i(t) \;=\; \sum_{n=1}^{\tau(t)-1} \big[v_{s_{n+1}}(t-1) - v_{s_n}(t-1)\big]\, e_i(n) + \big[r(t) - v_{s_{\tau(t)}}(t-1)\big]\, e_i(\tau(t)),$$

where $e_i(n)$ is the value of the eligibility trace for state i at step n. The explicit dependence of s_n and $e_i(n)$ on t, the trial number, is dropped for improved readability. At the beginning of each trial, the eligibility trace is zero for all states. It is updated for the three different algorithms as follows (also see Figure 1):

accumulate TD:

$$e_i(n) = \begin{cases} \lambda e_i(n-1) + 1 & \text{if } i = s_n, \\ \lambda e_i(n-1) & \text{if } i \neq s_n; \end{cases}$$

replace TD:

$$e_i(n) = \begin{cases} 1 & \text{if } i = s_n, \\ \lambda e_i(n-1) & \text{if } i \neq s_n; \end{cases}$$

first TD:

$$e_i(n+1) = \begin{cases} 1 & \text{if } i = s_n \text{ and this is the first visit to } i \text{ in this trial}, \\ \lambda e_i(n-1) & \text{else.} \end{cases}$$

In the appendix we present the above three TD algorithms in a different form that is more suited to the MSE calculations but is less intuitive because it does not separate out the calculation of the eligibility trace from the calculation of the δs.

There are interesting relationships between the MC and TD algorithms (Singh & Sutton, 1996; Barto & Duff, 1994) and among the different TD algorithms: every-visit MC is identical to accumulate TD(1), first-visit MC is identical to replace TD(1), accumulate TD(0) is identical to replace TD(0), and first TD(1) is identical to replace TD(1). Therefore for small values of λ, accumulate TD and replace TD are similar, while for large values of λ, replace TD and first TD are similar. This is reflected in the learning curves presented below (e.g., Figures 7 and 8).

All of the above MC and TD algorithms are known to converge asymptotically to $\mathbf{v}^*$ with probability one under the following conditions: a) $\alpha(t)$ decreases to 0 in an appropriate way, b) every state is visited infinitely often, and c) lookup tables are used to store the estimated value function.[4] In this paper we are less interested in asymptotic convergence than we are in the MSE performance in the shortterm under conditions of fixed or time varying $\alpha(t)$ and $\lambda(t)$.

3. Analytical Bias, Variance, and MSE Update Equations

This section provides equations that compute $\mathbf{b}(t)$, $C(t)$, and hence MSE(t), after trial t, based on the values of these same quantities at the start of the trial and as a

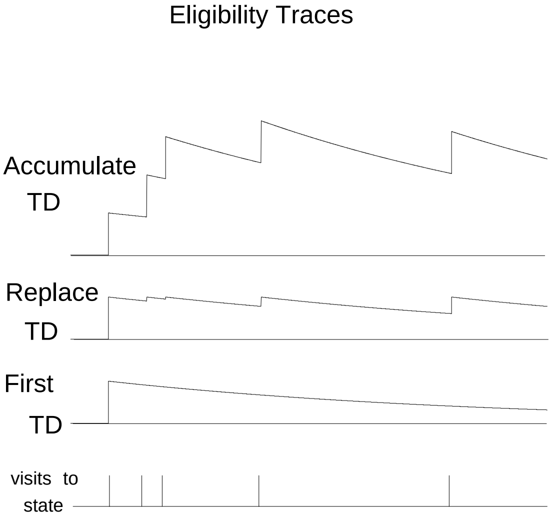

Figure 1. Three Different Eligibility Traces. In accumulate TD, each visit adds another eligibility trace to the previous trace. In replace TD, each visit to a state terminates the previous eligibility trace and initiates another trace. In first TD, only the first visit to a state in a trajectory initiates an eligibility trace.

function of the algorithm and the problem and their parameters. Instead of working directly with the bias $\mathbf{b}(t)$ and covariance $C(t)$ of the estimate $\mathbf{v}(t)$, we work with the mean $\mathbf{m}(t) = E\{\mathbf{v}(t)\}$, and the mean square matrix $S(t) = E\{\mathbf{v}(t)\mathbf{v}^T(t)\}$. Clearly, $\mathbf{b}(t) = \mathbf{v}^* - \mathbf{m}(t)$, and $C(t) = S(t) - \mathbf{m}(t)\mathbf{m}^T(t)$. To preserve readability, only the form of the final update equations are presented in this section (see the appendix for details).

The mean update equations of all the above algorithms take the form:

$$m_i(t) \;=\; m_i(t-1) + \alpha(t)\Gamma_i(t), \tag{5}$$

and the S updates take the form:

$$S_{ij}(t) = S_{ij}(t-1) + \alpha(t)\Delta_{ij}(t) + \alpha(t)^2\Upsilon_{ij}(t), \tag{6}$$

where $\Gamma(t)$, $\Delta(t)$ and $\Upsilon(t)$ depend on $\mathbf{m}(t-1)$ (and $\Delta(t)$ and $\Upsilon(t)$ depend on $S(t-1)$), differ for the different algorithms, and are distinguished when necessary by adding superscripts: FV for first-visit MC, EV for every-visit MC, A for accumulate TD, F for first TD, and R for replace TD. Throughout this paper use of these quantities without superscripts in an equation implies that it holds for all the algorithms with the appropriate superscripts appended. $\Gamma^{FV}, \Delta^{FV}, \Upsilon^{FV}$ are defined in Section A.1; $\Gamma^{EV}, \Delta^{EV}, \Upsilon^{EV}$ are defined in Section A.2; $\Gamma^{A}, \Delta^{A}, \Upsilon^{A}$ are defined in Section A.3; $\Gamma^{F}, \Delta^{F}, \Upsilon^{F}$ are defined in Section A.4; and $\Gamma^{R}, \Delta^{R}, \Upsilon^{R}$ are defined in Section A.5. The details of the S update equation take a considerable amount of space and, unfortunately, do not lead us to any direct conclusions about the effect of different parameters. The effect of the step-size, α, however, is clear from Equations 5 and 6: the bias update depends linearly on the step-size, while the covariance update has both linear and quadratic dependence on the step-size.

Given the update equations for $\mathbf{m}(t)$ and $S(t)$, the update equation for MSE is derived as follows:

$$\begin{aligned}
\text{MSE}(t) &= \sum_{i\in\mathbf{s}} p_i(b_i^2(t) + C_{ii}(t)) \\
&= \sum_{i\in\mathbf{s}} p_i((v_i^* - m_i(t))^2 + (S_{ii}(t) - m_i^2(t))) \\
&= \sum_{i\in\mathbf{s}} p_i(v_i^{*2} - 2v_i^* m_i(t) + S_{ii}(t)) \\
&= \sum_{i\in\mathbf{s}} p_i(v_i^{*2} - 2v_i^*(m_i(t-1) + \alpha(t)\Gamma_i(t)) \\
&\qquad\qquad + (S_{ii}(t-1) + \alpha(t)\Delta_{ii}(t) + \alpha^2(t)\Upsilon_{ii}(t))) \\
&= \sum_{i\in\mathbf{s}} p_i(v_i^{*2} - 2v_i^* m_i(t-1) + S_{ii}(t-1)) \\
&\quad + \alpha(t)\sum_{i\in\mathbf{s}} p_i(-2v_i^*\Gamma_i(t) + \Delta_{ii}(t)) + \alpha^2(t)\sum_{i\in\mathbf{s}} p_i\Upsilon_{ii}(t) \\
&= \text{MSE}(t-1) + \alpha(t)\sum_{i\in\mathbf{s}} p_i(-2v_i^*\Gamma_i(t) + \Delta_{ii}(t)) \\
&\qquad\qquad + \alpha^2(t)\sum_{i\in\mathbf{s}} p_i\Upsilon_{ii}(t).
\end{aligned} \tag{7}$$

4. Learning Curves on Specific Markov Reward Processes

We coded the analytical MSE update equations in the `C` programming language to develop a software analysis tool that, for a fixed Markov reward process, computes exact MSE curves for L trials in $O(|\mathbf{s}|^3 L)$ steps regardless of the behavior of the variance and bias curves. The analysis tool is simple to use. It takes as input the transition probability matrix and the mean and the variance of the terminal rewards of any Markov reward process that satisfies the assumptions of Section 2, the initial bias vector and covariance matrix (null, if the initial value function is fixed), a choice for α, λ, and the number of trials. Its output is a sequence of exact MSE values, one for each trial. Our software is available from ftp://ftp.cs.colorado.edu/users/baveja/AMse.tar.gz via anonymous ftp.

We applied our software to two classes of problems: a symmetric random walk (SRW; Figure 2), and a Markov reward process with a cyclicity parameter that controls the expected length of a trial by controlling the expected number of revisits to each non-terminal state (Figure 3). We use the first problem to explore the space of possible learning curve behaviors, the effect of increasing step-sizes, increasing λs, the relative performance of the three TD algorithms, and the relative performance of TD and MC algorithms. The latter problem is used to explore the effect of initial bias and chain cyclicity on optimal schedules of α and λ for the three TD algorithms.

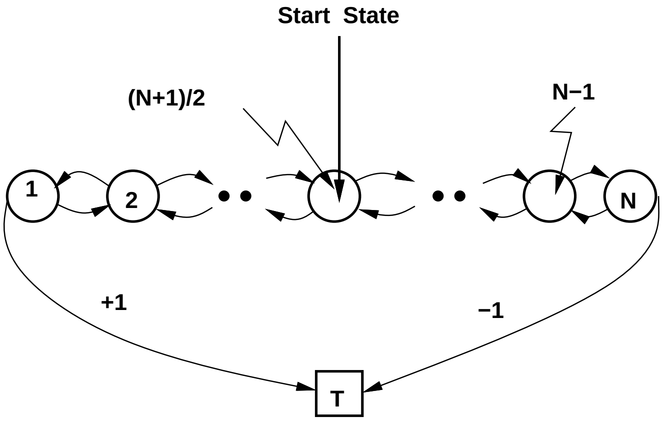

Figure 2. Symmetric Random Walk (SRW) Problem. The number of non-terminal states, N, is an odd number. T is the terminal state. In each non-terminal state there is equal probability of a transition to the left or to the right. Absorption from the left-end of the process rewards $+1$ while absorption from the right-end rewards -1. All other rewards are zero. All trials start in the middle state.

4.1. Analytical and Empirical MSE Curves

First, we present empirical confirmation of our analytical equations by comparing analytical and empirical MSE curves on the 19 state SRW problem. Empirical MSE curves average a number of sample MSE curves obtained through simulation runs. A simulation run sets a seed for the random number generator and then performs a specified number of trials. Different seeds are used for different simulation runs. Figure 4a shows analytical MSE curves for the three TD algorithms (see Figure 4 caption for details about α and λ). Figure 4b shows the difference between the analytical curves and the empirical curves produced by averaging more than three million simulation runs. The match after three million simulation runs was within four decimal places for all three algorithms.

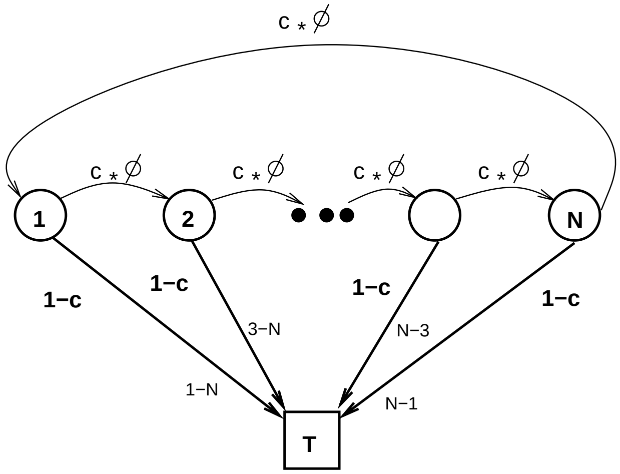

Figure 3. Parameterised Markov Reward Process. There are N non-terminal states labeled $1, \ldots, N$. T is the terminal state. The parameters c and ϕ together control the cyclicity of the Markov reward process. The closer the product $c * \phi$ is to one, the higher the cyclicity. For each state i, the remaining transition probability, $c - \phi * c$, is distributed equally among all other transitions (not shown here) out of state i. The reward for terminating from state i is $2i - N - 1$, and there is equal probability of starting in any non-terminal state.

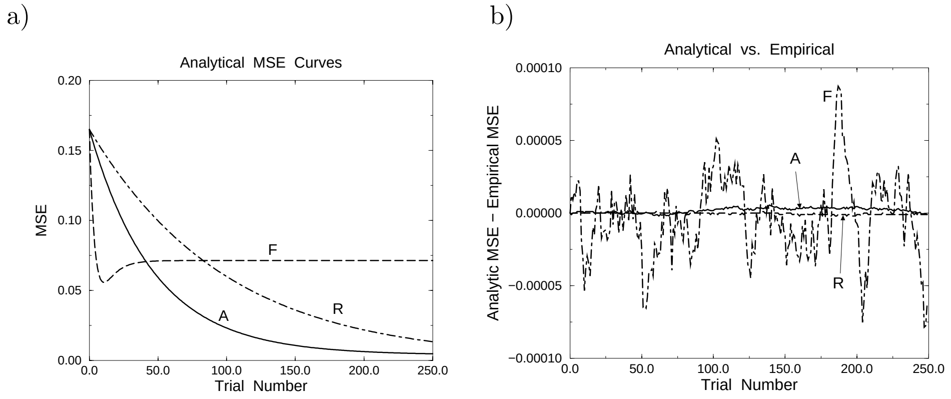

Figure 4. Comparing Analytical and Empirical MSE Curves. a) Analytical learning curves obtained on the 19 state SRW problem with parameters $\alpha = 0.01$, and $\lambda = 0.9$ for accumulate TD and replace TD, and $\alpha = 0.6$, and $\lambda = 0.9$ for first TD. b) The difference between the analytically-obtained MSE curves and the empirically obtained MSE curves. Values for λ and α were chosen to produce both monotonic and non-monotonic MSE curves. The empirical curves were obtained by averaging more than three million simulation runs. For each algorithm, the analytical and empirical MSE curves agree up to the fourth decimal place.

4.2. Long Tail Behavior in Empirical MSE Curves

In Figure 5 we present a case showing that the empirical simulation method for approximating MSE curves does not work well for some parameters for the algorithms. Figure 5a compares the analytical MSE curve with the empirical MSE curve obtained from more than 12 million simulation runs on a small five-state SRW problem. The algorithm parameters were chosen such that the asymptotic variance was high. The poor match and the spikiness of the empirical learning curve are explained by Figure 5b, which shows the empirical MSE after 198 trials as a function of the number of simulation runs averaged into the empirical MSE estimate. The sharp jump in the plot close to 6.5 million simulation runs is strong evidence of the long tails of the distribution of estimated values for these parameter choices. Figure 5c plots the distribution of the sample MSE values at trial 198. The inset graph shows that very large values of MSE occasionally occur. The mean MSE over 15.5 million trials is 0.3133, the variance over these trials is 9950.9 (standard error is 2.529). Straightforward averaging of samples from such distributions is known to be very slow to converge to the mean.[5]

The above demonstration that the distribution of estimated values can have a long tail underscores the need for caution in interpreting comparisons of algorithms based on empirical MSE curves, particularly results that compare algorithms over a wide range of algorithm parameters. Unfortunately, our analysis is unable to distinguish between the circumstances under which high asymptotic variance implies long tails and the circumstances under which it does not, for we found instances of both cases. In addition, the long tail of the distribution of estimated values does not explain the apparent low 'underlying' asymptote in the empirical MSE curve of Figure 5a.

4.3. Effect of $\boldsymbol{\alpha}$ and $\boldsymbol{\lambda}$ on TD Algorithms

In this section we study the effect of α and λ on TD algorithms. Figure 6 presents examples of the different kinds of bias, variance, and MSE (the sum of bias-square and variance) curves that are obtained from the 19 state SRW problem for fixed α and λ. Figure 6(a) and Figure 6(b) show examples of learning curves in which bias and variance both converge and in which bias converges while variance diverges. Figure 6c shows a case where both the bias and the variance diverge in accumulate TD. Figure 6d shows a case where both the bias and the variance converge in first TD. There are four classes of MSE curve behavior that result from the different combinations of bias and variance curve behavior: monotonically decreasing MSE that asymptotes to a non-zero value (e.g., replace TD in Figure 6a); first decreasing and then increasing MSE that asymptotes to a non-zero value (e.g., first TD in Figure 6d); and MSE first decreasing and then increasing to infinity (e.g., replace TD in Figure 6b). A fourth behavior in which the bias starts off so near to 0 that the MSE increases monotonically, is rarer.

In Figures 7 and 8 we summarize the effect of varying α and λ in the 19 state SRW problem. Each graph of Figure 7 plots MSE curves for a single constant λ and for all $\alpha \in \{0.001, 0.01, 0.075, 0.1, 0.6\}$. Each row corresponds to a different

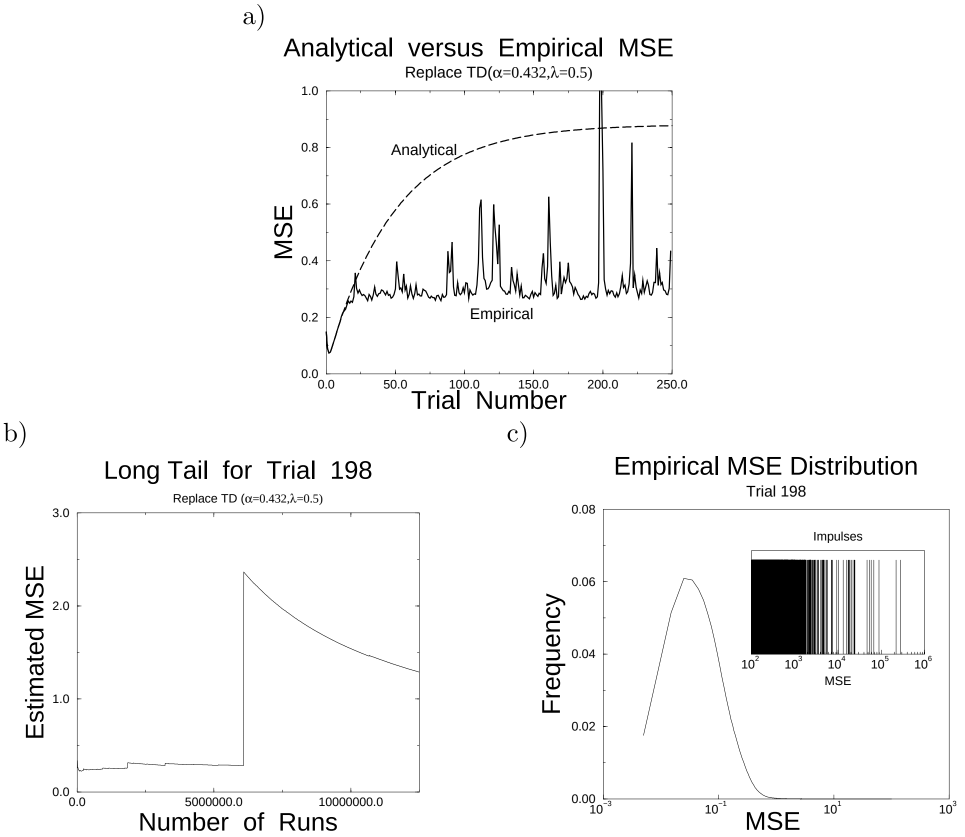

Figure 5. Long Tails of the Distribution of Estimated Values. a) A case in which the empirical method badly failed to match the analytical learning curve after more than 12 million simulation runs on a small 5 state SRW problem for parameters $\alpha = 0.432$ and $\lambda = 0.5$. The empirical learning curve is also very spiky. The real problem is illustrated in (b), which plots the estimated MSE on trial 198 as a function of the number of runs averaged to form the estimate. The big impulse around 6.5 million runs implies that within $10,000$ runs the MSE was large enough to take the average from 0.3 to 2.4. This implies that the distributions of the estimated values can have very long tails making the straight averaging method very slow. c) Empirical MSE data for the estimate at trial 198. The main graph shows the empirical distribution over 15.5 million simulation runs (based on a different set of seeds for the random number generator than for (a) and (b)). The inset shows impulses at actual sample values greater than 100. The largest value is greater than 200000.

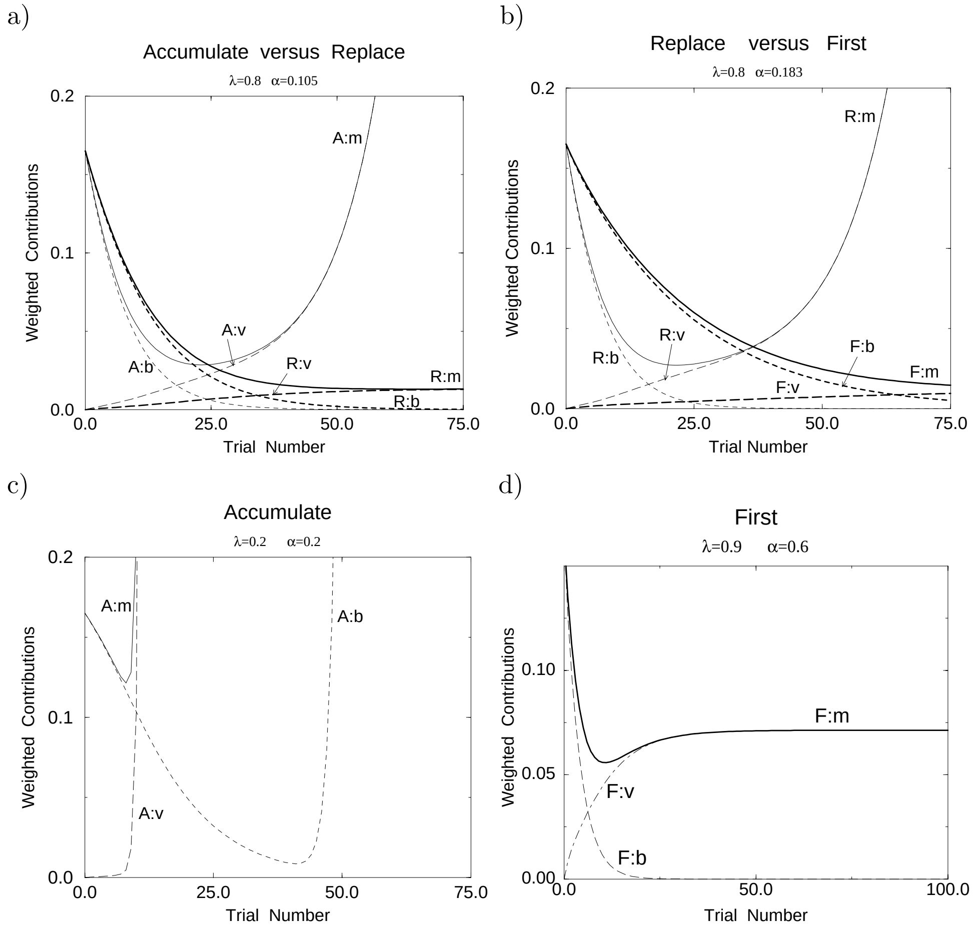

Figure 6. Different Kinds of Bias-Square, Variance and MSE Learning Curves (from the 19 state SRW problem). In all panels the labels A:b, R:b, and F:b, when present, denote the bias-square curve for accumulate TD, replace TD and first TD respectively, the labels A:v, R:v, and F:v denote the variance curve for accumulate TD, replace TD, and first TD respectively, and the labels A:m, R:m, and F:m denote the MSE curve for accumulate TD, replace TD and first TD respectively. (a,b) Examples of two cases: the bias and variance both converge, and the bias converges while the variance diverges. (c) Both the bias and the variance diverge, the bias more slowly than the variance. (d) An MSE curve with an interesting knee, or local minimum. In each panel, the MSE curve is the sum of the weighted bias-square and the weighted variance curves.

$\lambda \in \{0.0, 0.5, 0.9, 1.0\}$, while the different columns correspond to different algorithms. Figure 8 presents similar data, except that each graph plots MSE for all $\lambda \in \{0.0, 0.2, 0.6, 0.8, 0.9, 1.0\}$ and a single constant α. The initial value function was 0.0 for all graphs.

We define the maximal *feasible* α for a given λ to be the largest value such that the MSE has a finite asymptote. For graphical convenience, all the graphs in Figures 7 and 8 have the same upper limit on MSE, and so it is not always clear for some values of λ and α whether the MSE diverges or whether it converges to a value greater than 0.2. We address this explicitly in Figure 16.

The following summary hypotheses for TD algorithms can be formulated from the data shown in Figures 7 and 8:

H1 For a fixed Markov reward process and a constant λ, increasing α has two general effects on the learning curve: there is a largest value of α below which the bias converges to zero and above which the bias diverges (Sutton, 1988; Dayan, 1992), and there is a largest value of α below which the variance converges to a non-zero value and above which it diverges. These largest *feasible* values of α need not be the same for bias and variance. Based on our limited investigation of learning curves, we conjecture that the largest feasible value of α for bias is greater than or equal to the corresponding value for variance (Figure 9).

H2 For each algorithm, increasing α while holding λ fixed increases the asymptotic value of MSE. This is most clearly seen in the graphs for $\lambda = 0.9$ (Figure 7g,h,i) for all three algorithms. Similarly, increasing λ in the feasible range while holding α fixed increases the asymptotic value of MSE. This is most clearly seen in the graphs for $\alpha = 0.075$ (Figure 8g,h,i) for all three algorithms. Therefore, the smaller the constant α and λ, the smaller the asymptotic MSE.

H3 For each algorithm, larger values of α or λ lead to faster convergence to the asymptotic value of MSE if there exists one. Examples of this are seen in the $\lambda = 0.9$ graphs of Figure 7 and the $\alpha = 0.075$ graphs of Figure 8. This may break down for λ very near to 1.

H4 In general, for each algorithm as one decreases λ, the feasible range of α shrinks, i.e., larger α can be used with larger λ without causing excessive MSE. We explore this issue in Section 5.1 and Figure 16.

An apparent effect of varying λ and α in Figures 7 and 8 is the increasing stability as one moves from accumulate TD to replace TD and from replace TD to first TD. For the same small value of λ, larger values of α are feasible for replace TD compared with accumulate TD and for first TD compared with replace TD. This is also seen in Figure 6a,b where for the same λ and α, accumulate TD diverges while replace TD converges, and for another λ and α, replace TD diverges while first TD converges. However, note that the magnitude of the update in value function in all three TD algorithms depends on both α and the magnitude of the eligibility trace. The eligibility trace should in general be larger for accumulate TD than for replace TD, and larger for replace TD than for first TD, and this may account for the effect entirely. A rescaling of α in Figures 7 and 8 to take the maximum possible

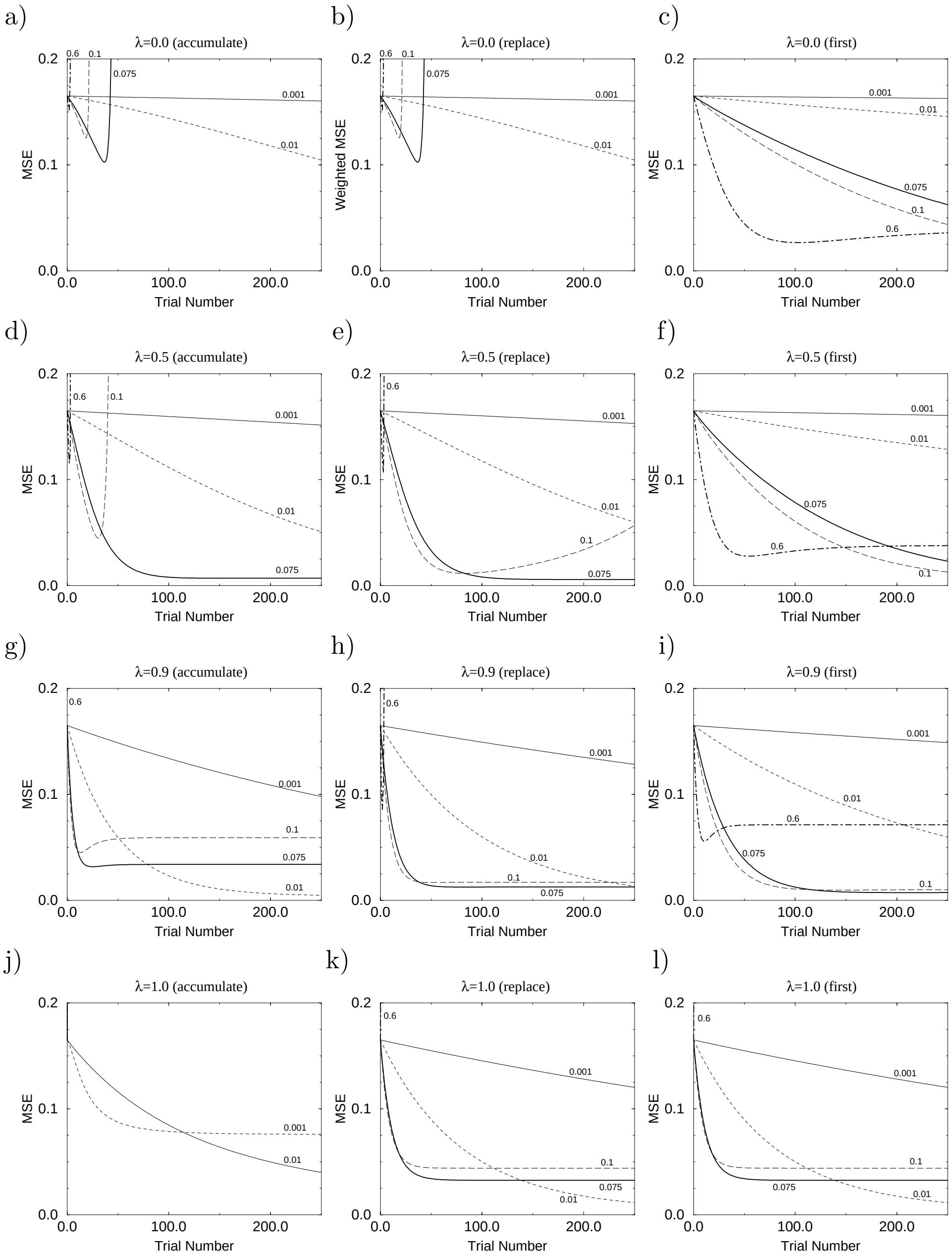

Figure 7. MSE Curves for Different Values of λ and α. The first column is for accumulate TD, the second for replace TD, and the third for first TD. Each row contains graphs for the same value of λ, with the λs increasing as we go down the columns. Each curve is for the given α. Note that for each column, as we increase λ, larger values of α become feasible (stable). For graphical convenience, all the graphs in Figures 7 and 8 have the same upper limit on MSE, and so it is not always clear for some values of λ and α whether the MSE diverges or whether it converges to a value greater than 0.2.

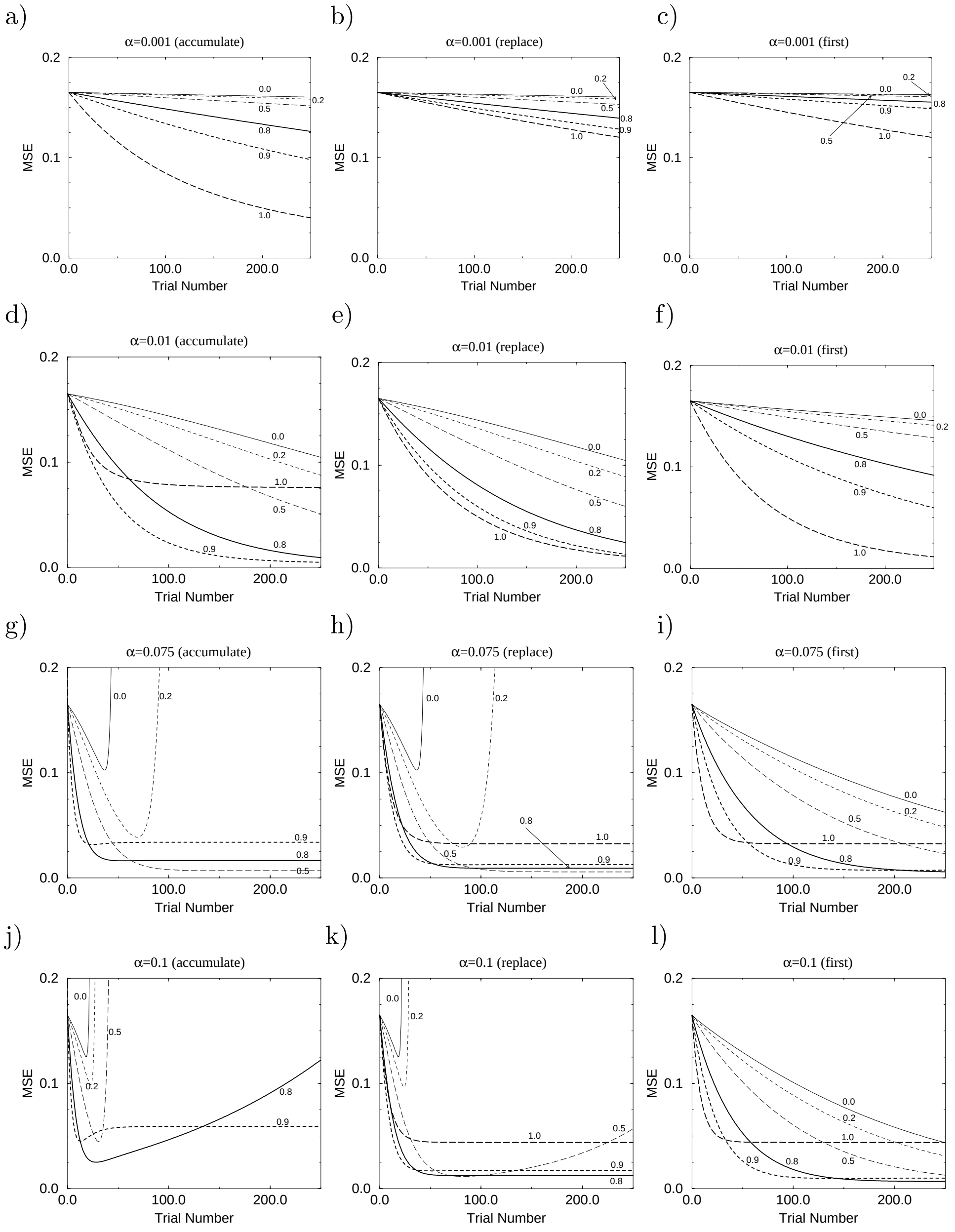

Figure 8. MSE Curves for Different Values of λ and α. Each panel is for a fixed α, and the individual curves are generated using the given value of λ. MSE curves for larger values of α and λ asymptote in fewer trials to larger asymptotic values. For graphical convenience, all the graphs in Figures 7 and 8 have the same upper limit on MSE, and so it is not always clear for some values of λ and α whether the MSE diverges or whether it converges to a value greater than 0.2.

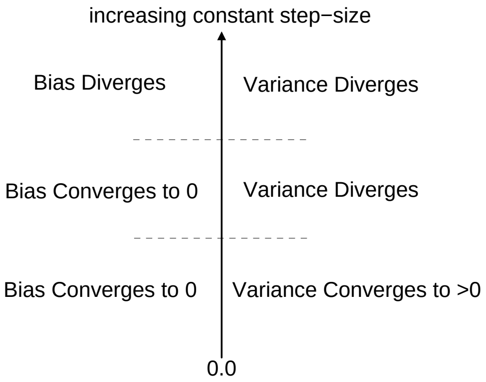

Figure 9. A Conjecture on Increasing Step-Sizes and Convergence in MC and TD Algorithms. It is known that there exists a small enough α below which the bias and variance converge to zero and a non-zero value respectively. It is also trivial to find a large enough α beyond which both the bias and variance diverge. The conjecture is that the largest feasible α for the bias is greater than or equal to the largest feasible α for the variance. Our admittedly limited empirical experience supports this result (see Figure 6 for an example.). Note that these critical values of α depend on the Markov reward process.

magnitude of eligibility traces into account may be appropriate (Sutton, personal communication). The greatest resulting difference would be for values of λ near $\lambda = 1$.

4.4. *One-step Optimal $\boldsymbol{\alpha}$ and $\boldsymbol{\lambda}$*

An advantage of having the analytical forms of the equations for the update of the mean and variance is that it is possible to optimize schedules for setting α and λ. Choosing the optimal schedules is useful in eliminating the effect of the choice of α when studying the effect of the λ parameter and vice versa. It is also useful in determining how problem parameters such as cyclicity and initial bias should affect our choice of α and λ schedules, and in determining whether one of the algorithms is to be preferred.

One-step Optimal Schedule for α

Given a particular λ, the effect on the MSE of a single step for any of the algorithms is quadratic in α. It is therefore straightforward to calculate the value of α that minimizes MSE after the next trial t, which we denote $\alpha_g(t)$:

$$\alpha_g(t) \;=\; \frac{\sum_{i\in\mathbf{s}} p_i(2v_i^*\Gamma_i(t) - \Delta_{ii}(t)))}{2.0\sum_{i\in\mathbf{s}} p_i\Upsilon_{ii}(t)}.$$

This is called the one-step optimal, or *greedy*, value of α. It is not clear that if one were interested in minimizing $\mathrm{MSE}(t+t')$, one would choose successive $\alpha(t), \alpha(t+1); \ldots$ that greedily minimize $\mathrm{MSE}(t), \mathrm{MSE}(t+1), \ldots$. In general, one could use our formulæ and dynamic programming to optimize a whole schedule for α, but this is computationally challenging.

Note that this technique for setting greedy α assumes complete knowledge of the Markov reward process and the initial bias and covariance of $\mathbf{v}(0)$, and is therefore not directly applicable to realistic applications of reinforcement learning.

One-step Optimal Schedule for λ

Calculating analytically the λ that would minimize $\mathrm{MSE}(t)$ given the bias and variance at trial $t-1$, which we denote $\lambda_g(t)$, is substantially harder than calculating $\alpha_g(t)$ because terms such as $(I-\lambda D)^{-1}$ for various matrices D enter Equation 7 when the details are filled in from the appendix. However, given any choice of λ, it is possible to compute the corresponding $\mathrm{MSE}(t)$. Therefore, we compute the one-step optimal, or *greedy*, value of λ to a desired accuracy by searching over appropriately-spaced λ-values between zero and one for the λ that yields minimum MSE. This is possible only because $\mathrm{MSE}(t)$ can be computed very cheaply using our analytical equations. The caveats about greediness in choosing $\alpha_g(t)$ also apply to $\lambda_g(t)$.

4.5. Performance as a Function of $\boldsymbol{\lambda}$

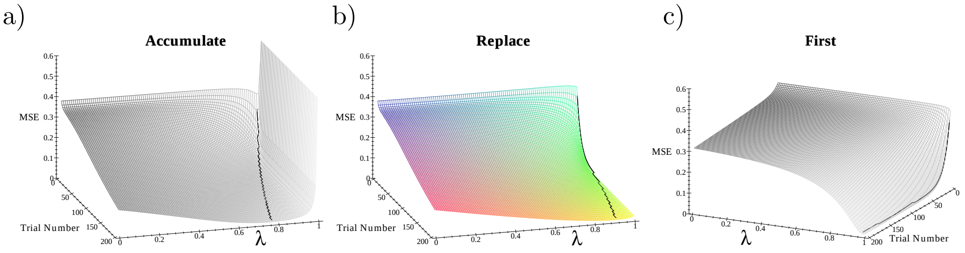

Figure 10. MSE Curves as a Function of λ. This figure plots the MSE as a function of both λ and trial number for $\alpha = 0.05$. For each trial number, the value of λ that achieves the minimum MSE is shown as a black line superimposed on the surface plot. These plots show that the minimum-error λ is not constant as a function of trial number, and that it generally shifts from a high initial value to lower values with increasing trial number. Decreasing the initial bias^2 would lower the initial best λs. Note that c) has a different rotation of the trial-number × λ plane.

Sutton (1988) and others have investigated the effect of λ on the empirical MSE at small trial numbers. The effect is usually summarized by U-shaped curves of empirical MSE at trial N as a function of λ. These curves provide evidence of the utility of eligibility traces, because $\lambda > 0$ gives minimum error, and also of the utility of TD over MC, because the minimum error λ is strictly less than one. We plot similar graphs here using our analytical MSE curves, except that we are also interested in the value of the minimum error λ as a function of trial number.

Figure 10 plots the MSE as a function of both λ and trial number for $\alpha = 0.05$. Note that in each panel of Figure 10, slices corresponding to fixed trial numbers are U-shaped. For each trial number, the value of λ that achieves the minimum MSE is shown as a black line superimposed on the surface plot. These plots show that the minimum-error λ is not constant as a function of trial number, and that it generally shifts from a high initial value to lower values with increasing trial number. Because larger values of λ converge to their asymptote faster (H3), for small trial number they tend to be winners in the race for smaller MSE. Values of λ that are too large, on the other hand, lead to rapid divergence. This explains the U-shaped curves for a fixed trial number as in Figure 10. Furthermore, because the asymptotes are smaller for smaller λ (H2), smaller values of λ tend to win for larger t. This may account for the decreasing value of the minimum-error λ as a function of t. However, this is all for α's that do not vary with trial number.

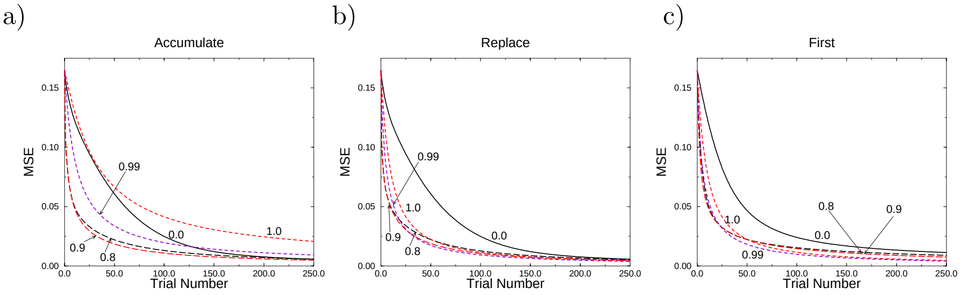

Figure 11. Greedy α Curves. These figures plot MSE for various values of λ using greedy (one-step optimal) step-sizes. The minimum-error λ starts high but then moves towards smaller values with increasing trial number.

We observe the same effects when α is allowed to vary with trial number, t. Ideally, one should search over all possible α schedules. Instead, for computational convenience, Figure 11 plots the MSE for $\lambda \in \{0.0, 0.8, 0.9, 0.99, 1.0\}$ using greedy α schedules. It is clear from this figure that no one λ dominates for all trial numbers. Further, more evidence is seen for U-shaped MSE curves as a function of λ at a fixed trial number by considering the MSE values at specific trial numbers in Figure 11. For example, in accumulate TD, $\lambda = 0.8$ has the smallest MSE for small t, for larger t, $\lambda = 0.9$ has the smallest MSE and then finally near the end $\lambda = 0.0$ has the smallest MSE. Similar effects are present in the replace TD and first TD graphs. Sutton's (personal communication) point made above that $1/(1-\lambda)$ might be a more reasonable scale than λ also applies to the discussion in this section.

Our results provide additional evidence for, and suggest an explanation for, the advantage of intermediate values of λ. However, we should note at least two reasons to be cautious about such empirical evidence presented by picking an arbitrary stopping point, especially based on a small trial number: 1) the MSE for the minimum (λ, α) pair so determined may actually diverge for larger trial numbers, and 2) if the variance of the value function is high at the stopping trial, then empirical

MSE values obtained from averaging even a very large number of simulated trials may be very inaccurate (e.g., Figure 5). We show below in Figure 13 that in fact the drop in MSE may be very insensitive to the value of λ except in the very first few trials, given the ability to schedule α appropriately.

4.6. Effect of Cyclicity and Initial Bias

In this section we consider a small 5 state process of the kind shown in Figure 3. The goal is to study the effect of varying cyclicity and initial bias on greedy λ and α schedules. The four rows of Figure 12 correspond to the four combinations of high and low values of both cyclicity and initial bias. The first column plots the MSE curves for all the algorithms, the second plots the greedy λ schedules, while the third plots the greedy α schedules. These results suggests the following conjecture (Sutton, 1988; Watkins, 1989) about the relationship between initial bias and greedy λ:

H5 If the initial value function has a high bias, one should begin with a large λ, while if the initial value function has a low bias, one should begin with a small λ. Over time the effect of the initial bias weakens and the asymptotic λ should depend mainly on other problem parameters.

With a large λ all three algorithms put greater trust in the payoff data than in the estimated values of intervening states. Conversely, with a smaller λ the estimated values of states are trusted more than the payoff data. Therefore, hypothesis H5 is intuitively reasonable because with a high initial bias, estimated values should be trusted less than payoff data. Similarly, with a low initial bias estimated values are close to correct and therefore should be trusted more than noisy payoff data. Clear evidence for hypothesis H5 is seen in Figure 12. The first and third rows correspond to high initial bias, and in both cases the initial λs are close to one. Rows two and four correspond to low initial bias and have low initial λs. We observe that the λ values after 75 trials are nearly the same if the amount of cyclicity is the same. The sharp jump of the λ value for first TD in Figure 12e is explained below.

Further evidence for hypotheses H3 and H4 is also seen in Figure 12. We suspect from hypothesis H3 that larger values of α lead to faster convergence to the associated asymptote and so one should want to use large αs, at least in the beginning. However, H4 suggests that the largest feasible α is larger for larger λ. Accordingly, we see high initial αs in rows 1 and 3 of Figure 12 that have high initial λs, and we see low initial αs in rows 2 and 4 that have lower initial λs.

The effect of cyclicity on the different algorithms is less clear. Increasing cyclicity should lead to more revisits to states before termination and should therefore amplify the relative differences between accumulate TD and replace TD, as well as between replace TD and first TD. However, from the results in Figure 12a,d,g,j it seems that by choosing the α and λ schedules wisely, the differences between the algorithms largely disappear. Of course, in practice the knowledge required to choose optimal, or even greedy, α and λ is not available and so for practical choices of α and λ, the differences may be more prominent (e.g., Singh & Sutton, 1996). Higher

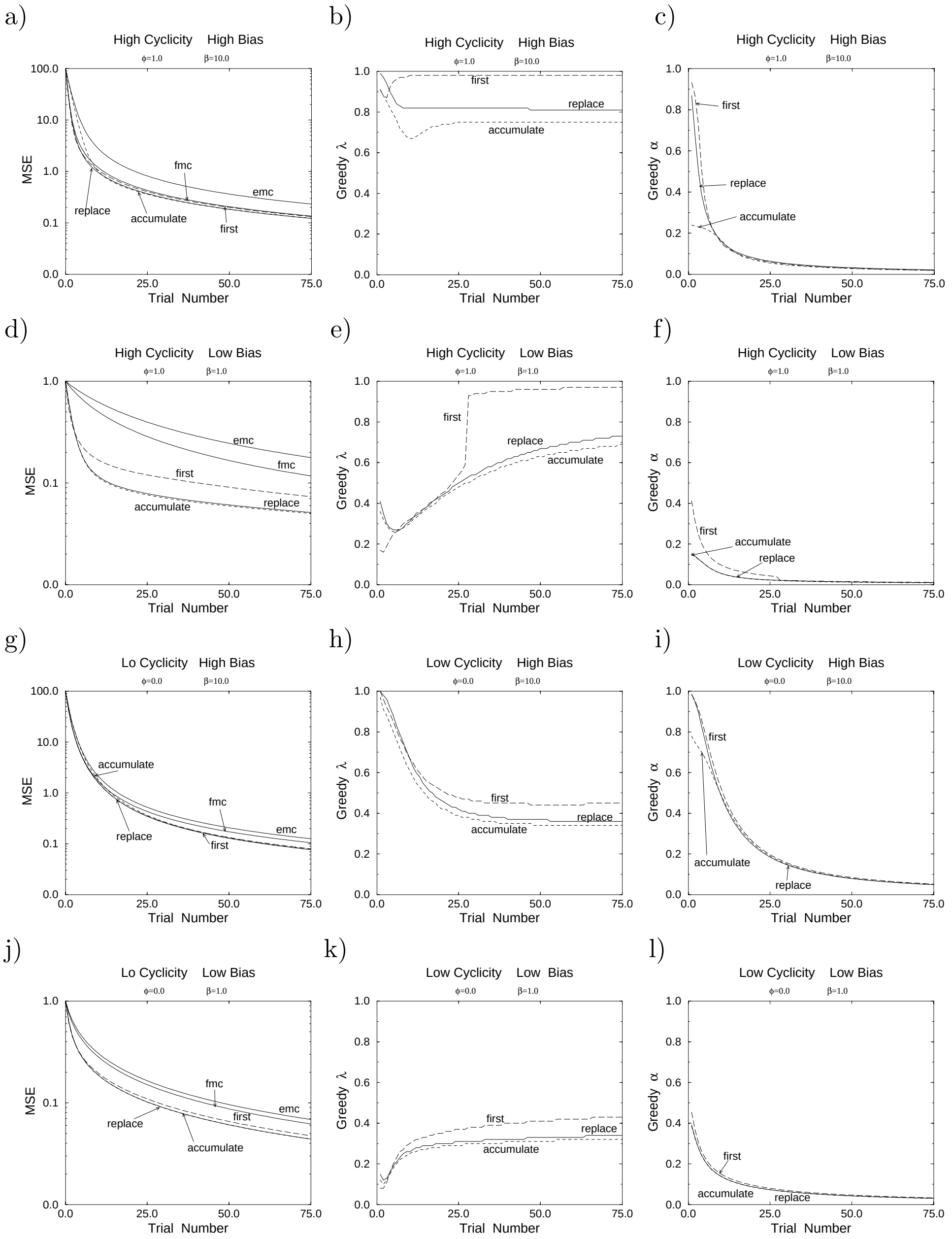

Figure 12. Effect of Problem Parameters. These figures show the behavior of the algorithms on a 5 state Markov reward process whose cyclicity (probability of revisits) is controlled by parameters ϕ and c. The parameter c was fixed to 0.9. Initial bias (β) was controlled separately. Greedy choices of α and λ were used. High initial bias leads to high initial λ. High cyclicity leads to high asymptotic values of λ. MSE curves for first-visit MC (fmc) and every-visit MC (emc) are also shown. In (j), the curves for replace TD and accumulate TD are indistinguishable.

cyclicity also resulted in larger asymptotic λ_g (compare rows 1 and 2 of Figure 12 with rows 3 and 4), because it leads to longer trials and therefore requires larger λ to obtain the same mix of the random payoff in the estimator than it would with shorter trials.

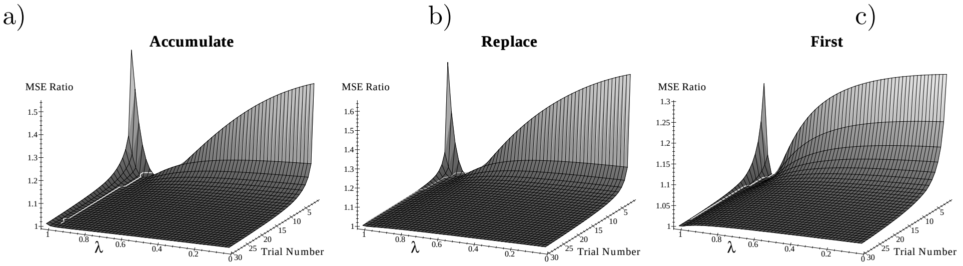

Figure 13. Sensitivity to λ. These surfaces show the ratios between the one-step MSE for all values of λ and the one-step MSE for the one-step optimal value of λ (all values of the ratios are ≥ 1). Each value of λ uses its own greedy value of α. On each successive trial, the best λ is chosen and then the MSEs are calculated for the next trial starting from the bias-square and variance that results from this choice. The white lines mark the best λ. These ratios are all close to 1 for trial numbers greater than 10.

But how sensitive is the rate of convergence to the choice of λ? Figure 13 suggests that careful choice of this parameter is only rewarded very near the beginning, and that over time the drop in MSE is relatively insensitive to choice of λ. Figure 13 plots the sensitivity to λ as a function of trial number. We measure sensitivity as the ratio of the resulting MSE when λ is used instead of λ_g. The step-size used is the greedy α associated with each λ. A white line is superimposed on the surface plot to mark the λ_g schedule. All three algorithms start out by being very sensitive to the choice of λ but soon the surface becomes very flat. This helps explain the sudden jump in λ_g in Figure 12e.

4.7. Comparing Algorithms

The first column of Figure 12 also compares the performance of the two MC algorithms with the three TD algorithms. In all cases, first-visit MC performs better than every-visit MC, and this is consistent with Singh & Sutton's (1996) theoretical results. In all cases, TD algorithms performed better than, or at least no worse than, MC algorithms. The difference between the MC and TD curves becomes small if the initial few greedy λ_g are close to 1, for in such cases there is little difference between MC and TD algorithms. Figure 14 compares the performances of MC and TD algorithms on the 5 state SRW problem. Figure 14 also plots the empirical MSE curve for the maximum-likelihood (ML) algorithm. The ML algorithm uses the trials to build a maximum-likelihood model of the transition probabilities and the rewards. Its estimate after n trials is the value function that would be correct if its estimated model after n trials were correct. The ML algorithm is computation-

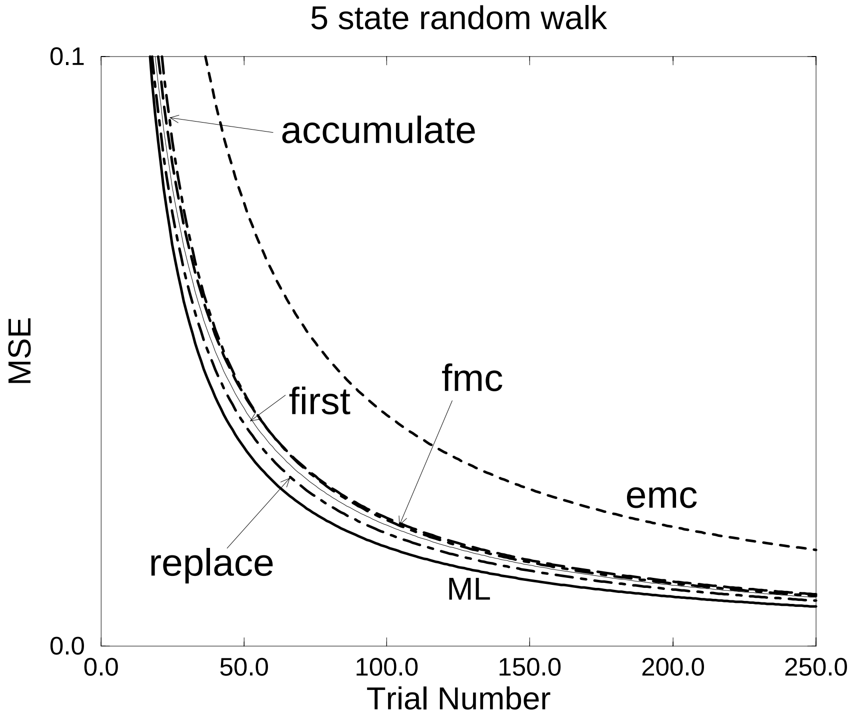

Figure 14. Comparing TD, MC and ML Algorithms. Comparison of the TD algorithms with the MC algorithms and the Maximum-Likelihood (ML) algorithm on the five state SRW problem. For the TD algorithms the greedy α and λ schedules were used. For the first-visit MC (fmc) and every-visit MC (emc) algorithms the greedy α schedules were used. The ML empirical MSE curve was obtained by averaging 19 million simulation runs.

ally very expensive for large problems and is therefore of interest only as an ideal to compare against. As expected, it forms a lower bound to all the other MSE curves.

5. Analysis of Asymptotic Convergence Rates

Given the analytical forms of the equations for the update of the mean, **m**, and the mean square matrix, S, it is possible, for fixed λ and α, to compute the asymptotic rates of convergence for **m** and S. To do so we rewrite Equations 5 and 6 in the following form:

$$\mathbf{m}(t) = \boldsymbol{a}^m + B^m \mathbf{m}(t-1) \quad (8)$$

$$S(t) = A^S + B^S S(t-1) + D^S \mathbf{m}(t-1), \quad (9)$$

where B^m depends linearly on α, and A^S, B^S and D^S depend at most quadratically on α.

The maximum moduli of the eigenvalues of B^m and B^S determine the fact and speed of convergence of the algorithms to finite endpoints. If either is greater than 1, then the algorithms will not converge in general. As illustrated below, we observed that the mean update is more stable than the mean-square update, i.e., the larger values of α still lead to eigenvalues of B^m that satisfy the convergence criteria.

Further, we know that for α sufficiently small, the mean converges to $\mathbf{v}^*$, and therefore we can determine the asymptotic $S(\infty)$ as:

$$S(\infty) = \left[\mathcal{I} - B^S\right]^{-1} \left[A^S + D^S \mathbf{v}^*\right]. \tag{10}$$

This formula is only true, of course, if the eigenvalues of B^S are less than 1 in modulus. We can calculate the value of α at which this ceases being true, a value we call the largest feasible α.

Just like the LMS algorithm (Widrow & Stearns, 1985), these algorithms converge at best with probability 1 to an ϵ-ball around $\mathbf{v}^*$ for a constant finite step-size. This amounts to the MSE converging to a fixed value which is determined by Equation 10. One can therefore use Equation 10 to determine which values of α lead to which terminal MSE, and, by calculating the eigenvalues of B^m, one can determine an upper bound to the rate of decrease of the error in the mean of the estimate.

5.1. Eigenvalue Analysis

We applied this eigenvalue analysis to accumulate TD on the 19 state SRW problem. Figure 15a shows the smallest and largest eigenvalue of the matrix B^m which governs the convergence of the bias to 0. The eigenvalues are real since the problem is symmetric. The smaller the moduli of these eigenvalues, the faster the mean can be guaranteed to converge. We observe that the bias reduces fastest for $\lambda = 1$.

Figure 15b shows the equivalent reduction rates for the matrix B^S, which governs the convergence of the mean square S. These maximal rates are only valid once the bias has converged to 0. However, we have always observed that the bias converges more rapidly than the mean square, at least if either converges. The algorithm diverges if the reduction rate is greater than one. For $\alpha = 0.075$, the smallest value of λ that ensures that the mean square converges is approximately 0.3, and is shown as limiting the region of instability in Figure 15a.

Figure 15c combines eigenvalue analysis for the mean with terminal MSE analysis from the mean square. For a given λ and α, we can solve Equation 9 with $\mathbf{m} = \mathbf{v}^*$ to calculate the terminal S and consequently the terminal MSE. We used numerical methods to find the step-size, α, that would give particular terminal MSE, and then found, for this α, the largest eigenvalue of the mean update matrix B^m. For some values of λ, there may be no α that gives a convergent S for a given MSE – indeed this is apparent in the graph. We show the consequent maximal mean reduction rate as a function of λ for two different terminal MSEs. Obviously, the more lax one is about the terminal MSE, the faster the convergence can be expected to be. Note that using an intermediate value of $\lambda < 1$ is optimal even though for any fixed value of α, Figure 15a implies that the larger λ the better. The explanation comes from Figure 16b, which shows the terminal MSE as a function of both λ and α. It is apparent that setting λ very near to 1 means that very small values of α must be used, thus reducing the maximal mean reduction rate.

Figure 16a shows the (numerically calculated) largest value of α for which S does not diverge. Note the kink in the curve for λ near 1 (amplified in the inset) which is a reason why larger values of λ are not always better. Figure 16b's plot of the terminal MSE as a function of α and λ shows that it is not only the largest feasible α that is important, but also the terminal MSE that results. This also has anomalous behavior as $\lambda \to 1$.

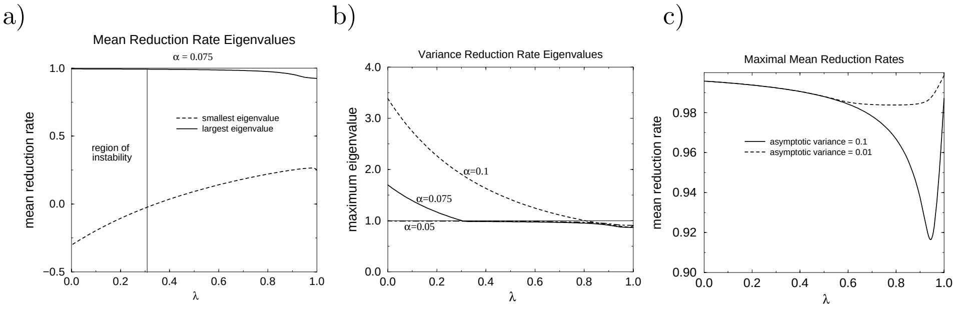

Figure 15. Eigenvalue Analyses of Bias and Mean Square Reduction. All three graphs are for the 19 state SRW and accumulate TD. a) Maximal and minimal eigenvalues of the bias update matrix as a function of λ for $\alpha = 0.075$. The mean square update is divergent for λ in the region of instability. b) Maximal modulus of the eigenvalues for the mean square update matrix for three values of α. Values greater than 1 lead to instability. c) Maximal modulus of the eigenvalues for the bias update matrix as a function of λ where α is chosen so that the terminal MSE is less than or equal to 0.1 or 0.01. Note that $\lambda = 1$ is not optimal.

6. Conclusions

We have provided analytical expressions for calculating how the bias and variance of various TD and Monte Carlo algorithms change over iterations. The expressions themselves seem not to be very revealing, but we provided many illustrations of their behavior in some particular Markov reward processes. We have also used the analysis to calculate one-step optimal (greedy) values of the step-size, α, and the eligibility-trace parameter, λ. Using these values makes the algorithms quite similar. Further, we calculated terminal mean square errors and maximal bias reduction rates.

Since all these results depend on the precise Markov reward processes chosen, it is hard to make generalizations. We have nevertheless posited four broad conjectures:

- for constant λ, the larger α, the larger the terminal MSE;

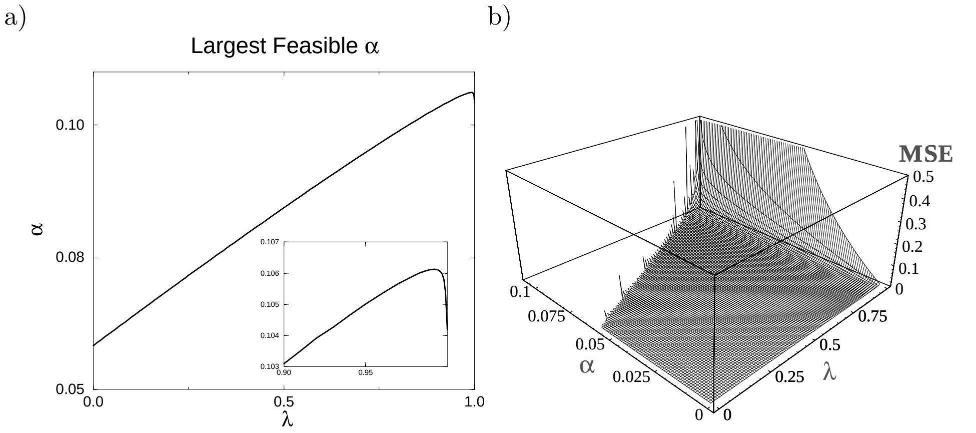

Figure 16. Feasible αs. Both graphs are for the 19 state random walk and accumulate TD. a) The largest value of α such that the MSE does not diverge. These were calculated numerically by finding the points as in Figure 15 where the mean square reduction rates cross the value 1. b) Terminal MSE as a function of α and λ. These are calculated using the mean square update matrix. The jaggedness comes from the relatively sharp cut-off to divergence.

- the larger α or λ (except for λ very close to 1), the faster the convergence to the asymptotic MSE, provided that this is finite;
- the smaller λ, the *smaller* the range of α for which the terminal MSE is not excessive;
- higher values of λ are good for cases with high initial biases.

The third of these is somewhat surprising because the effective value of the step-size is really $\alpha/(1-\lambda)$ and so one would expect to be able to use *larger* α as λ gets further from 1. However, the lower λ, the more the value of a state is based on the value estimates for nearby states. We conjecture that with small λ, large α can quickly lead to high correlation in the value estimates of nearby states and result in runaway variance updates. However, with larger λ, larger α stay feasible in part because of the larger influence of both farther away (and hence less correlated in value estimates) states and particularly because of the uncorrelated payoffs. We saw evidence for a side-effect of this in Figure 12 where higher cyclicity led to higher asymptotic λ_g because it compounds the problem of dependence on nearby states for small λ.

Two issues require comment: the role of λ and the relative merits of the algorithms that we studied. Two main lines of evidence suggest that using values of λ other than 1 (i.e., using a temporal difference rather than a Monte-Carlo algorithm) can be beneficial. First, the *greedy* value of λ chosen to minimize the MSE at the end of the step (whilst using the associated greedy α) remains away from 1 (see Figure 12). Interestingly, it remains away from 0 also. As the bias tends to 0, one might expect that the greedy λ would tend to 0 too, since the smaller the (fixed) λ, the smaller the asymptotic MSE. However, the smaller the λ, the lower the feasible step-size α, and so the less the one-step reduction in the MSE. The curves in Figure 12 suggest that the greedy value of λ converges to a value intermediate between 0 and 1 with the number of trials, but this conclusion is not supported by any analysis. In any event, in this limit, the differences between different values of λ are extremely small (as shown in Figure 13). Note that the greedy value of α tends slowly to 0, as one might expect. The second piece of evidence favoring $\lambda \neq 1$ comes from the eigenvalue analysis in Figures 15 and 16. For fixed α, the terminal variance is higher for $\lambda = 1$; the largest value of α that can be used is higher for $\lambda < 1$; and the asymptotic speed with which the bias can be guaranteed to decrease fastest is higher for $\lambda < 1$.

We had expected that there would be large differences between the three different TD algorithms: accumulate TD, replace TD and first TD. Singh & Sutton (1996) analyzed slightly different versions of accumulate TD and replace TD for $\lambda = 1$, showing that the MSE of accumulate TD is lower at the start of learning, but becomes higher than that of replace TD after some number of trials. However, our results show that given suitable choices of α and λ, the algorithms are essentially indistinguishable – we have cases in which accumulate TD does better, worse, or the same as replace TD. Of course, we used complete knowledge of the Markov reward process to calculate the appropriate parameters, and we have not addressed the sensitivity of the MSE to inappropriate choices.

This analysis clearly provides only an early step toward understanding the course of learning for TD algorithms, and has focused exclusively on prediction rather than control. The analytical expressions for MSE might lend themselves to general conclusions over whole classes of Markov reward processes. In addition, it would be useful to understand the conditions leading to the apparent long tails in Figure 5 and to the convergence of greedy values of λ in Figure 12.

Acknowledgments

We thank Rich Sutton and Andy Barto for their painstaking reading of this paper and their many comments that have improved it. Leslie Kaelbling, Michael Kearns, Michael Jordan, Lawrence Saul, Tommi Jaakkola and Rob Schapire provided valuable discussions and comments at various stages of this work and we thank them, as well as Michael Kearns for impressing us along this path. We also thank the anonymous reviewers for many useful comments. Part of this research was done while SS was a Postdoctoral fellow at MIT with Professor Michael Jordan, where he was supported by grants from ATR Human Information Processing Research and from Siemens Corporation. PD was supported by MIT.

Appendix

MSE Calculations

The three TD algorithms can be defined without separating out the eligibility trace calculations (as in Section 2.1). However, we will need additional notation: $s_m(t); m \geq 1$ is the state at step m of trial t, $\tau(t)$ is the number of steps in trial t, and $n_i(t;d)$ is the step in trial t at which the d^{th} visit to state i occurs. $K_i(t;n)$ is one if state i is visited at step n of trial t, and is zero otherwise. If a trial lasts k steps then it results in a sequence of k states followed by a payoff. Hereafter, whenever it leads to no ambiguity, we drop the explicit dependence of various quantities on the trial number t.

accumulate TD:

$$\begin{aligned} v_i(t) &= v_i(t-1) + \alpha(t)\Big(\sum_{n=1}^{\tau(t)} K_i(t;n)\Big[\sum_{m=n+1}^{\tau(t)} (1-\lambda)\lambda^{m-n-1} v_{s_m}(t-1) \\ &\qquad + \lambda^{\tau(t)-n} r(t)\Big] - \kappa_i(t) v_i(t-1)\Big) \end{aligned}$$

replace TD:

$$\begin{aligned} v_i(t) &= v_i(t-1) + \alpha(t)\Big(\ \Big[\sum_{d=1}^{\kappa_i(t)-1} \sum_{m=n_i(t;d)+1}^{n_i(t;d+1)} (1-\lambda)\lambda^{m-n_i(t;d)-1} v_{s_m}(t-1)\Big] \\ &\qquad + \Big[\sum_{m=n_i(t;\kappa_i(t))+1}^{\tau(t)} (1-\lambda)\lambda^{m-n_i(t;\kappa_i(t))-1} v_{s_m}(t-1)\Big] \\ &\qquad + \lambda^{\tau(t)-n_i(t;\kappa_i(t))} r(t) - \kappa_i(t) v_i(t-1)\Big) \end{aligned}$$

first TD:

$$\begin{aligned} v_i(t) &= v_i(t-1) + \alpha(t)\Big(\ \Big[\sum_{m=n_i(t;1)+1}^{\tau(t)} (1-\lambda)\lambda^{m-n_i(t;1)-1} v_{s_m}(t-1)\Big] \\ &\qquad + \lambda^{\tau(t)-n_i(t;1)} r(t) - K_i(t) v_i(t-1)\Big) \end{aligned}$$

As in the main text, consider absorbing Markov reward processes with state set $\mathbf{s}$, with only terminal payoffs, and offline updating. We repeat the definitions of several basic quantities in Table A.1 and define other useful symbols that serve as labels for often repeated pieces of formulæ in Table A.2. Below, δ_{ij} is the Kronecker delta function, and $\otimes$ denotes the element wise product. To enhance readability, we drop the dependence of $\mathbf{m}$ and S on trial number $t-1$ on the right hand sides of most equations below.

Table A.1. Definitions Revisited

transition matrix for non-terminals	Q
probability of termination from i	q_i
reward for terminating from i	r_i
variance of the reward from i	h_i^2
random value function after trial t	$\mathbf{v}(t)$
step-size for trial t	$\alpha(t)$
trace parameter for trial t	$\lambda(t)$
mean value function after trial t	$\mathbf{m}(t) = E\{\mathbf{v}(t)\}$
mean squared value function after trial t	$S_{ij}(t) = E\{v_i(t)v_j(t)\}$
covariance of value function after trial t	$C_{ij}(t) = S_{ij}(t) - m_i(t)m_j(t)$
bias of value function after trial t	$b_i(t) = v_i^* - v_i(t)$
squared error of value function	$\epsilon_i(t) = b_i^2(t) + C_{ii}(t)$

A.1. Bias & Covariance calculations for first-visit MC

The mean of the value function gets updated as follows:

$$m_i(t) \;=\; m_i(t-1) + \alpha(t)\Gamma_i^{FV}(t),$$

where

$$\Gamma_i^{FV}(t) \;=\; (\mu^T[I + [[Q_{-i}][I-Q]^{-1}]])_i(v_i^* - m_i(t-1)) \tag{A.1}$$

The S update is as follows:

$$S_{ij}(t) \;=\; S_{ij}(t-1) + \alpha(t)\Delta_{ij}^{FV}(t) + \alpha(t)^2\Upsilon_{ij}^{FV}(t), \tag{A.2}$$

where

$$\begin{aligned}\Delta_{ij}^{FV}(t) \;=\; & 2\delta_{ij}\Big((\mu^T[I + [[Q_{-i}][I-Q]^{-1}]])_i(v_i^* m_i - S_{ii})\Big) \\ & +(1.0 - \delta_{ij})\Big((\mu^T[I + [[Q_{-i}][I-Q]^{-1}]])_j(v_j^* m_i - S_{ij}) \\ & +(\mu^T[I + [[Q_{-i}][I-Q]^{-1}]])_i(v_i^* m_j - S_{ij})\Big), \text{ and}\end{aligned} \tag{A.3}$$

$$\begin{aligned}\Upsilon_{ij}^{FV}(t) \;=\; & \delta_{ij}\Big((\mu^T[I + [[Q_{-i}][I-Q]^{-1}]])_i(S_{ii} + ([I-Q]^{-1}\mathbf{r2})_i - 2v_i^* m_i)\Big) \\ & +KS_{ij}(S_{ij} + ([I-Q]^{-1}\mathbf{r2})_j - m_i v_j^* - m_j v_j^*) \\ & +KS_{ji}(S_{ji} + ([I-Q]^{-1}\mathbf{r2})_i - m_j v_i^* - m_i v_i^*)\end{aligned} \tag{A.4}$$

(A.5)

Table A.2. Some Useful Intermediate Quantities

transition matrix with i^{th} row set to 0	Q_{-i}
transition matrix with i^{th} and j^{th} rows set to 0	$Q_{-i,-j}$
expected one-step payoff; ($\mathbf{r1}$)	$\mathbf{r1} = \mathbf{q} \otimes \mathbf{r}$
expected one-step squared payoff; ($\mathbf{r2}$)	$\mathbf{r2} = \mathbf{q} \otimes (\mathbf{r} \otimes \mathbf{r} + \mathbf{h}^2)$
true value function; ($\mathbf{v}^*$)	$\mathbf{v}^* = [I - Q]^{-1}\mathbf{r1}$
exp. number of visits to state i; n_i	$\mathbf{n}^T = \mu^T[I - Q]^{-1}$
Distribution over states in a trial	$p_i = \frac{n_i}{\sum_j n_j}$
exp. number of visits to j without visiting i	$(D_{-i})_j = (\mu^T[I - Q_{-i}]^{-1})_j$
exp. number of visits to k without visiting i, j	$(D_{-i,-j})_k = (\mu^T[I - Q_{-i,-j}]^{-1})_k$
for i, $j \in \mathbf{s}$	$DD_{i,j} = (D_{-i,-j})_i[Q_{-j}[I - Q_{-j}]^{-1}]_{ij}$ $+(D_{-j,-i})_j[Q_{-i}[I - Q_{-i}]^{-1}]_{ji}$
for $i \in \mathbf{s}$	$K_i = (1.0 - \lambda(t))(Q[I - \lambda(t)Q^{-1}]\mathbf{m})_i$ $+([I - \lambda(t)Q]^{-1}\mathbf{r1})_i$
for i, $j \in \mathbf{s}$	$KV_{ij} = (1.0 - \lambda(t))[Q[I - \lambda(t)Q^{-1}]S]_{ij}$ $+((([I - \lambda(t)Q]^{-1}\mathbf{r1})_i m_j$
for i, $j \in \mathbf{s}$	$KS_{ij} = (\mu^T[I + [Q_{-i,-j}][I - Q_{-i,-j}]^{-1}])_i$ $\times[[Q_{-j}][I - Q_{-j}]^{-1}]]_{ij}$
Weighted mean squared error after trial t	$\text{MSE}(t) = \sum_{i \in \mathbf{s}} p_i \epsilon_i(t)$

A.2. Bias & Covariance Calculations for every-visit MC

The mean of the value function gets updated as follows:

$$m_i(t) \;=\; m_i(t-1) + \alpha(t)\Gamma_i^{EV}(t),$$

where

$$\Gamma_i^{EV}(t) \;=\; n_i(v_i^* - m_i(t-1)). \tag{A.6}$$

The S update is as follows:

$$S_{ij}(t) = S_{ij}(t-1) + \alpha(t)\Delta_{ij}^{EV}(t) + \alpha(t)^2\Upsilon_{ij}^{EV}(t), \tag{A.7}$$

where

$$\begin{aligned} \Delta_{ij}^{EV}(t) \;=\;& \delta_{ij}[2n_i(m_i v_i^* - S_{ii})] \\ & +(1.0 - \delta_{ij})[n_i m_j + n_j m_i - m_i n_j v_j^* - m_j n_i v_i^*], \text{ and} \end{aligned} \tag{A.8}$$

$$\begin{aligned}\Upsilon^{EV}_{ij}(t) &= \delta_{ij}[(2n_i[Q[I-Q]^{-1}]_{ii})(([I-Q]^{-1}\mathbf{r2})_i - 2v^*_i m_i + S_{ii})] \\ &\quad +(1.0-\delta_{ij})\Big((n_i[Q[I-Q]^{-1}]_{ij}([I-Q]^{-1}\mathbf{r2})_j \\ &\quad +n_j[Q[I-Q]^{-1}]_{ji}([I-Q]^{-1}\mathbf{r2})_i) \\ &\quad -((m_i+m_j)(n_i[Q[I-Q]^{-1}]_{ij}v^*_j + n_j[Q[I-Q]^{-1}]_{ji}v^*_i)) \\ &\quad +S_{ij}(n_i[Q[I-Q]^{-1}]_{ij} + n_j[Q[I-Q]^{-1}]_{ji})\Big)\end{aligned} \tag{A.9}$$

A.3. Bias & Covariance Calculations for accumulate TD

The mean of the value function gets updated as follows:

$$m_i(t) = m_i(t-1) + \alpha(t)\Gamma^A_i(t),$$

where

$$\begin{aligned}\Gamma^A_i(t) &= n_i\Big((1.0-\lambda(t))([Q][I-\lambda(t)[Q]]^{-1}\mathbf{m}(t-1))_i \\ &\quad +([I-\lambda(t)Q]^{-1}\mathbf{r1})_i - m_i(t-1)\Big)\end{aligned} \tag{A.10}$$

The S update is as follows:

$$S_{ij}(t) = S_{ij}(t-1) + \alpha(t)\Delta^A_{ij}(t) + \alpha(t)^2\Upsilon^A_{ij}(t), \tag{A.11}$$

where

$$\Delta^A_{ij}(t) = n_i(KV_{ij} - S_{ij}(t-1)) + n_j(KV_{ji} - S_{ji}(t-1)), \tag{A.12}$$

$$\begin{aligned}\Upsilon^A_{ij}(t) &= S_{ij}(n_i[Q[I-Q]^{-1}]_{ij} + n_j[Q[I-Q]^{-1}]_{ji}) \\ &\quad +\delta_{ij}n_iS_{ii} \\ &\quad -n_j[Q[I-Q]^{-1}]_{ji}KV_{ij} - (1.0-\lambda(t))n_i[Q[I-\lambda(t)Q]^{-1}]_{ij}S_{jj} \\ &\quad -\lambda(t)n_i[Q[I-\lambda(t)Q]^{-1}]_{ij}KV_{jj} \\ &\quad -\sum_{k\in\mathbf{s}}(1.0-\lambda(t))n_i[Q[I-\lambda(t)Q]^{-1}]_{ik}[Q[I-Q]^{-1}]_{kj}S_{kj} \\ &\quad -\delta_{ij}n_iKV_{ii} \\ &\quad -n_i[Q[I-Q]^{-1}]_{ij}KV_{ji} - (1.0-\lambda(t))n_j[Q[I-\lambda(t)Q]^{-1}]_{ji}S_{ii} \\ &\quad -\lambda(t)n_j[Q[I-\lambda(t)Q]^{-1}]_{ji}KV_{ii} \\ &\quad -\sum_{k\in\mathbf{s}}(1.0-\lambda(t))n_j[Q[I-\lambda(t)Q]^{-1}]_{jk}[Q[I-Q]^{-1}]_{ki}S_{ki} \\ &\quad -\delta_{ij}n_iKV_{ii} \\ &\quad +\sum_{k\in\mathbf{s}}\sum_{m\in\mathbf{s}}\Big((1.0-\lambda(t))^2n_i[Q[I-\lambda(t)Q]^{-1}]_{ik}[Q[I-Q]^{-1}]_{kj} \\ &\qquad\qquad [Q[I-\lambda(t)Q]^{-1}]_{jm}S_{km}\Big)\end{aligned}$$

$$
\begin{aligned}
&+\sum_{k\in\mathbf{s}}\Big((1.0-\lambda(t))n_i[Q[I-\lambda(t)Q]^{-1}]_{ik}[Q[I-Q]^{-1}]_{kj}\\
&\qquad\qquad([I-\lambda(t)Q]^{-1}\mathbf{r1})_j m_k\Big)\\
&+(1.0-\lambda(t))n_i[Q[I-\lambda(t)Q]^{-1}]_{ij}KV_{jj}\\
&+\sum_{k\in\mathbf{s}}(1.0-\lambda(t))^2 n_i\lambda(t)[Q[I-\lambda(t)Q]^{-1}]_{ij}[Q[I-\lambda(t)^2Q]^{-1}]_{jk}S_{kk}\\
&+n_i\lambda(t)[Q[I-\lambda(t)Q]^{-1}]_{ij}([I-\lambda(t)^2Q]^{-1}\mathbf{r2})_j\\
&+\sum_{k\in\mathbf{s}}\sum_{m\in\mathbf{s}}(1.0-\lambda(t))^2\lambda(t)n_i[Q[I-\lambda(t)Q]^{-1}]_{ij}\\
&\qquad\Big([Q[I-\lambda(t)^2Q]^{-1}]_{jk}\lambda(t)[Q[I-\lambda(t)Q]^{-1}]_{km}\\
&\qquad\qquad+[Q[I-\lambda(t)^2Q]^{-1}]_{jm}\lambda(t)[Q[I-\lambda(t)Q]^{-1}]_{mk}\Big)S_{mk}\\
&+\sum_{k\in\mathbf{s}}\Big(2.0(1.0-\lambda(t))n_i\lambda^2(t)[Q[I-\lambda(t)Q]^{-1}]_{ij}\\
&\qquad\qquad[Q[I-\lambda(t)^2Q]^{-1}]_{jk}([I-\lambda(t)Q]^{-1}\mathbf{r1})_k m_k\Big)\\
&+\sum_{k\in\mathbf{s}}\sum_{m\in\mathbf{s}}\Big((1.0-\lambda(t))^2 n_j[Q[I-\lambda(t)Q]^{-1}]_{jk}[Q[I-Q]^{-1}]_{ki}\\
&\qquad\qquad[Q[I-\lambda(t)Q]^{-1}]_{im}S_{km}\Big)\\
&+\sum_{k\in\mathbf{s}}\Big((1.0-\lambda(t))n_j[Q[I-\lambda(t)Q]^{-1}]_{jk}[Q[I-Q]^{-1}]_{ki}\\
&\qquad\qquad([I-\lambda(t)Q]^{-1}\mathbf{r1})_i m_k\Big)\\
&+(1.0-\lambda(t))n_j[Q[I-\lambda(t)Q]^{-1}]_{ji}KV_{ii}\\
&+\sum_{k\in\mathbf{s}}(1.0-\lambda(t))^2 n_j\lambda(t)[Q[I-\lambda(t)Q]^{-1}]_{ji}[Q[I-\lambda(t)^2Q]^{-1}]_{ik}S_{kk}\\
&+n_j\lambda(t)[Q[I-\lambda(t)Q]^{-1}]_{ji}([I-\lambda(t)^2Q]^{-1}\mathbf{r2})_i\\
&+\sum_{k\in\mathbf{s}}\sum_{m\in\mathbf{s}}(1.0-\lambda(t))^2\lambda(t)n_j[Q[I-\lambda(t)Q]^{-1}]_{ji}\\
&\qquad\Big([Q[I-\lambda(t)^2Q]^{-1}]_{ik}\lambda(t)[Q[I-\lambda(t)Q]^{-1}]_{km}\\
&\qquad\qquad+[Q[I-\lambda(t)^2Q]^{-1}]_{im}\lambda(t)[Q[I-\lambda(t)Q]^{-1}]_{mk}\Big)S_{km}\\
&+\sum_{k\in\mathbf{s}}\Big(2.0(1.0-\lambda(t))n_j\lambda^2(t)[Q[I-\lambda(t)Q]^{-1}]_{ji}[Q[I-\lambda(t)^2Q]^{-1}]_{ik}\\
&\qquad\qquad([I-\lambda(t)Q]^{-1}\mathbf{r1})_k m_k\Big)\\
&+\delta_{ij}\sum_{k\in\mathbf{s}}(1.0-\lambda(t))^2 n_i[Q[I-\lambda(t)^2Q]^{-1}]_{ik}S_{kk}
\end{aligned}
$$

$$
\begin{aligned}
&+\delta_{ij} n_i([I - \lambda(t)^2 Q]^{-1}\mathbf{r2})_i \\
&+\delta_{ij}\sum_{k\in\mathbf{s}}\sum_{m\in\mathbf{s}}(1.0-\lambda(t))^2 n_i\Big(\Big[[Q[I-\lambda(t)^2Q]^{-1}]_{ik} \\
&\qquad\qquad \lambda(t)[Q[I-\lambda(t)Q]^{-1}]_{km}\Big] \\
&\qquad\qquad +[Q[I-\lambda(t)^2Q]^{-1}]_{im}\lambda(t)[Q[I-\lambda(t)Q]^{-1}]_{mk}\Big)S_{mk} \\
&+\delta_{ij}\sum_{k\in\mathbf{s}}\Big(2.0(1.0-\lambda(t))n_i\lambda(t)[Q[I-\lambda(t)^2Q]^{-1}]_{ik} \\
&\qquad\qquad ([I-\lambda(t)Q]^{-1}\mathbf{r1})_k m_k\Big). \qquad \text{(A.13)}
\end{aligned}
$$

A.4. Bias & Covariance Calculations for first TD

The mean of the value function gets updated as follows:

$$m_i(t) \;=\; m_i(t-1) + \alpha(t)\Gamma_i^F(t),$$

where

$$\Gamma_i^F(t) \;=\; (D_{-i})_i(K_i - m_i(t-1)) \qquad \text{(A.14)}$$

The S update is as follows:

$$S_{ij}(t) = S_{ij}(t-1) + \alpha(t)\Delta_{ij}^F(t) + \alpha(t)^2\Upsilon_{ij}^F(t), \qquad \text{(A.15)}$$

where

$$\Delta_{ij}^F(t) \;=\; (D_{-i})_i(KV_{ij} - S_{ij}(t-1)) + (D_{-j})_j(KV_{ji} - S_{ji}(t-1)), \qquad \text{(A.16)}$$

and

$$
\begin{aligned}
\Upsilon_{ij}^F(t) \;=\; & DD_{i,j}S_{ij} \\
& -(1-\delta_{ij})(D_{-i,-j})_i[[Q_{-j}][I-Q_{-j}]^{-1}]_{ij}KV_{ji} \\
& -(1-\delta_{ij})\sum_{k\in\mathbf{s}}\Big((1.0-\lambda(t))(D_{-i,-j})_j[[Q_{-i}][I-\lambda(t)Q_{-i}]^{-1}]_{jk} \\
&\qquad\qquad [[Q_{-i}][I-Q_{-i}]^{-1}]_{ki}S_{ki}\Big) \\
& -(1-\delta_{ij})\Big((D_{-i,-j})_j[[Q_{-i}][I-\lambda(t)Q_{-i}]^{-1}]_{ji} \\
&\qquad\qquad ((1.0-\lambda(t))S_{ii} + \lambda(t)KV_{ii})\Big) \\
& -\delta_{ij}(D_{-i})_iKV_{ii} \\
& -(1-\delta_{ij})(D_{-j,-i})_j[[Q_{-i}][I-Q_{-i}]^{-1}]_{ji}KV_{ij} \\
& -(1-\delta_{ij})\sum_{k\in\mathbf{s}}\Big((1.0-\lambda(t))(D_{-j,-i})_i[[Q_{-j}][I-\lambda(t)Q_{-j}]^{-1}]_{ik}
\end{aligned}
$$

$$
\begin{aligned}
&\qquad\qquad [[Q_{-j}][I-Q_{-j}]^{-1}]_{kj}S_{kj}\Bigr)\\
&-(1-\delta_{ij})\Bigl((D_{-j,-i})_i[[Q_{-j}][I-\lambda(t)Q_{-j}]^{-1}]_{ij}\\
&\qquad\qquad ((1.0-\lambda(t))S_{jj}+\lambda(t)KV_{jj})\Bigr)\\
&-\delta_{ij}(D_{-i})_iKV_{ii}\\
&+(1-\delta_{ij})\sum_{km}\Bigl((1.0-\lambda(t))^2(D_{-i,-j})_i[[Q_{-j}][I-\lambda(t)Q_{-j}]^{-1}]_{ik}\\
&\qquad\qquad [[Q_{-j}][I-Q_{-j}]^{-1}]_{kj}[Q[I-\lambda(t)Q]^{-1}]_{jm}S_{km}\Bigr)\\
&+(1-\delta_{ij})\sum_{k}\Bigl((1.0-\lambda(t))(D_{-i,-j})_i[[Q_{-j}][I-\lambda(t)Q_{-j}]^{-1}]_{ik}\\
&\qquad\qquad [[Q_{-j}][I-Q_{-j}]^{-1}]_{kj}([I-\lambda(t)Q]^{-1}\mathbf{r1})_jm_k\Bigr)\\
&+(1-\delta_{ij})(1.0-\lambda(t))(D_{-i,-j})_i[[Q_{-j}][I-\lambda(t)Q_{-j}]^{-1}]_{ij}KV_{jj}\\
&+(1-\delta_{ij})\sum_{k}\Bigl((1.0-\lambda(t))^2(D_{-i,-j})_i\lambda(t)[[Q_{-j}][I-\lambda(t)Q_{-j}]^{-1}]_{ij}\\
&\qquad\qquad [Q[I-\lambda(t)^2Q]^{-1}]_{jk}S_{kk}\Bigr)\\
&+(1-\delta_{ij})\Bigl((D_{-i,-j})_i\lambda(t)[[Q_{-j}][I-\lambda(t)Q_{-j}]^{-1}]_{ij}\\
&\qquad\qquad ([I-\lambda(t)^2Q]^{-1}\mathbf{r2})_j\Bigr)\\
&+(1-\delta_{ij})\sum_{km}(1.0-\lambda(t))^2\lambda(t)(D_{-i,-j})_i[[Q_{-j}][I-\lambda(t)Q_{-j}]^{-1}]_{ij}\\
&\qquad\qquad \Bigl([Q[I-\lambda(t)^2Q]^{-1}]_{jk}\lambda(t)[Q[I-\lambda(t)Q]^{-1}]_{km}\\
&\qquad\qquad +[Q[I-\lambda(t)^2Q]^{-1}]_{jm}\lambda(t)[Q[I-\lambda(t)Q]^{-1}]_{mk}\Bigr)S_{mk}\\
&+(1-\delta_{ij})\sum_{k}2.0(1.0-\lambda(t))(D_{-i,-j})_i\lambda(t)[[Q_{-j}][I-\lambda(t)Q_{-j}]^{-1}]_{ij}\\
&\qquad\qquad \lambda(t)[Q[I-\lambda(t)^2Q]^{-1}]_{jk}([I-\lambda(t)Q]^{-1}\mathbf{r1})_km_k\\
&+(1-\delta_{ij})\sum_{km}(1.0-\lambda(t))^2(D_{-j,-i})_j[[Q_{-i}][I-\lambda(t)Q_{-i}]^{-1}]_{jk}\\
&\qquad\qquad [[Q_{-i}][I-Q_{-i}]^{-1}]_{ki}[Q[I-\lambda(t)Q]^{-1}]_{im}S_{km}\\
&+(1-\delta_{ij})\sum_{k}(1.0-\lambda(t))(D_{-j,-i})_j[[Q_{-i}][I-\lambda(t)Q_{-i}]^{-1}]_{jk}\\
&\qquad\qquad [[Q_{-i}][I-Q_{-i}]^{-1}]_{ki}([I-\lambda(t)Q]^{-1}\mathbf{r1})_im_k\\
&+(1-\delta_{ij})(1.0-\lambda(t))(D_{-j,-i})_j[[Q_{-i}][I-\lambda(t)Q_{-i}]^{-1}]_{ji}KV_{ii}\\
&+(1-\delta_{ij})\sum_{k}\Bigl((1.0-\lambda(t))^2(D_{-j,-i})_j\lambda(t)[[Q_{-i}][I-\lambda(t)Q_{-i}]^{-1}]_{ji}
\end{aligned}
$$

$$
\begin{aligned}
&\qquad\qquad [Q[I-\lambda(t)^2Q]^{-1}]_{ik}S_{kk}\Big) \\
&+(1-\delta_{ij})(D_{-j,-i})_j\lambda(t)[[Q_{-i}][I-\lambda(t)Q_{-i}]^{-1}]_{ji}([I-\lambda(t)^2Q]^{-1}\mathbf{r2})_i \\
&+(1-\delta_{ij})\sum_{km}(1.0-\lambda(t))^2\lambda(t)(D_{-j,-i})_j[[Q_{-i}][I-\lambda(t)Q_{-i}]^{-1}]_{ji} \\
&\qquad\qquad \Big([Q[I-\lambda(t)^2Q]^{-1}]_{ik}\lambda(t)[Q[I-\lambda(t)Q]^{-1}]_{km} \\
&\qquad\qquad [Q[I-\lambda(t)^2Q]^{-1}]_{im}\lambda(t)[Q[I-\lambda(t)Q]^{-1}]_{mk}\Big)S_{mk} \\
&+(1-\delta_{ij})\sum_{k}2(1.0-\lambda(t))(D_{-j,-i})_j\lambda(t)[[Q_{-i}][I-\lambda(t)Q_{-i}]^{-1}]_{ji} \\
&\qquad\qquad \lambda(t)[Q[I-\lambda(t)^2Q]^{-1}]_{ik}([I-\lambda(t)Q]^{-1}\mathbf{r1})_k m_k \\
&+\delta_{ij}\sum_{k}(1.0-\lambda(t))^2(D_{-i})_i[Q[I-\lambda(t)^2Q]^{-1}]_{ik}S_{kk} \\
&+\delta_{ij}(D_{-i})_i([I-\lambda(t)^2Q]^{-1}\mathbf{r2})_i \\
&+\delta_{ij}\sum_{km}(1.0-\lambda(t))^2(D_{-i})_i\Big([Q[I-\lambda(t)^2Q]^{-1}]_{ik} \\
&\qquad\qquad \lambda(t)[Q[I-\lambda(t)Q]^{-1}]_{km} \\
&\qquad\qquad +[Q[I-\lambda(t)^2Q]^{-1}]_{im}\lambda(t)[Q[I-\lambda(t)Q]^{-1}]_{mk}\Big)S_{km} \\
&+\delta_{ij}\sum_{k}\Big(2.0(1.0-\lambda(t))(D_{-i})_i\lambda(t)[Q[I-\lambda(t)^2Q]^{-1}]_{ik} \\
&\qquad\qquad ([I-\lambda(t)Q]^{-1}\mathbf{r1})_k m_k\Big).
\end{aligned}
\tag{A.17}
$$

A.5. Bias & Covariance Calculations for replace TD

We need to define some additional quantities here:

$$
\begin{aligned}
M_{ij} \;=\;& \frac{\Big((1.0-\lambda(t))\sum_{k\neq j}[Q[I-\lambda(t)Q_{-j}]^{-1}]_{jk}(S_{ki}(t-1)-S_{ji}(t-1))\Big)}{1-[Q[I-Q_{-j}]^{-1}]_{jj}} \\
&+\frac{\mathbf{r1}_j m_i(t-1)-q_jS_{ji}(t-1)}{1-[Q[I-Q_{-j}]^{-1}]_{jj}} \\
&+\frac{\Big(\lambda(t)\sum_{k\neq j}[Q[I-\lambda(t)Q_{-j}]^{-1}]_{jk}(\mathbf{r1}_k m_i(t-1)-q_kS_{ji}(t-1))\Big)}{1-[Q[I-Q_{-j}]^{-1}]_{jj}}
\end{aligned}
$$

The mean of the value function gets updated as follows:

$$
m_i(t) \;=\; m_i(t-1)+\alpha(t)\Gamma_i^R(t),
$$

where

$$
\Gamma_i^R(t) \;=\; (D_{-i})_i\Bigg(\frac{\sum_j[Q[I-\lambda(t)Q_{-i}]^{-1}]_{ij}(m_j(t-1)-m_i(t-1))}{1-[Q[I-Q_{-i}]^{-1}]_{ii}}
$$

$$
\begin{aligned}
&\left.\frac{(1.0-\lambda(t))}{1-[Q[I-Q_{-i}]^{-1}]_{ii}}\right) \\
&+(D_{-i})_i\left(\frac{\lambda(t)\sum_{j\neq i}[Q[I-\lambda(t)Q_{-i}]^{-1}]_{ij}(\mathbf{r1}_j-q_j m_i(t-1))}{1-[Q[I-Q_{-i}]^{-1}]_{ii}}\right. \\
&\qquad\left.+\frac{\mathbf{r1}_i-q_i m_i(t-1)}{1-[Q[I-Q_{-i}]^{-1}]_{ii}}\right). \qquad \text{(A.18)}
\end{aligned}
$$

The S update is as follows:

$$S_{ij}(t)=S_{ij}(t-1)+\alpha(t)\Delta^R_{ij}(t)+\alpha(t)^2\Upsilon^R_{ij}(t), \qquad \text{(A.19)}$$

where

$$\Delta^R_{ij}(t) \;=\; (D_{-j})_j M_{ij}+(D_{-i})_i M_{ji}. \qquad \text{(A.20)}$$

To define Υ^R, we need to compute the following intermediate quantities:

$$
\begin{aligned}
E_{ij}(t) \;=\; &\sum_{l\neq i,j}\sum_{k\neq i,j}(1.0-\lambda(t))^2\lambda(t)[[Q_{-i}][I-\lambda^2(t)Q_{-i,-j}]^{-1}]_{jl} \\
&\qquad [[Q_{-i,-j}][I-\lambda(t)Q_{-i,-j}]^{-1}]_{lk} \\
&\qquad (S_{k,l}(t)+S_{i,j}(t)-S_{i,l}(t)-S_{k,j}(t)) \\
&+\sum_{l\neq i,j}(1.0-\lambda(t))^2[[Q_{-i}][I-\lambda^2(t)Q_{-i,-j}]^{-1}]_{jl} \\
&\qquad (S_{l,l}(t)+S_{i,j}(t)-S_{i,l}(t)-S_{l,j}(t)) \\
&+\sum_{l\neq i,j}\sum_{k\neq j}(1.0-\lambda(t))^2\lambda(t)[[Q_{-i}][I-\lambda^2(t)Q_{-i,-j}]^{-1}]_{jl} \\
&\qquad [[Q_{-j}][I-\lambda(t)Q_{-j}]^{-1}]_{lk} \\
&\qquad (S_{l,k}(t)+S_{i,j}(t)-S_{i,k}(t)-S_{l,j}(t)) \\
&+\sum_{l\neq i,j}\sum_{k\neq j}(1.0-\lambda(t))\lambda(t)[[Q_{-i}][I-\lambda^2(t)Q_{-i,-j}]^{-1}]_{jl} \\
&\qquad [I-\lambda(t)Q_{-j}]^{-1}_{lk}\Big(\mathbf{r1}_k(m_l-m_i)-q_k(S_{jl}-S_{ij})\Big) \\
&+\sum_{l\neq i,j}(1.0-\lambda(t))[[Q_{-i}][I-\lambda(t)Q_{-i,-j}]^{-1}]_{jl}[[Q_{-j}][I-Q_{-j}]^{-1}]_{lj} \\
&\qquad (M_{lj}-M_{ij}) \\
&+\sum_{l\neq i,j}\sum_{k\neq i,j}(1.0-\lambda(t))\lambda(t)[[Q_{-i}][I-\lambda^2(t)Q_{-i,-j}]^{-1}]_{jl} \\
&\qquad [I-\lambda(t)Q_{-i,-j}]^{-1}_{lk}\Big(\mathbf{r1}_k(m_l-m_j)+q_k(S_{ij}-S_{il})\Big) \\
&+\sum_{l\neq i,j}\lambda^2(t)[[Q_{-i}][I-\lambda^2(t)Q_{-i,-j}]^{-1}]_{jl} \\
&\qquad \Big(\mathbf{r2}_l-\mathbf{r1}_l(m_j+m_i)+q_l S_{ij}\Big)
\end{aligned}
$$

$$
\begin{aligned}
&+\mathbf{r2}_j - \mathbf{r1}_j(m_j + m_i + q_j S_{ij} \\
&+ \sum_{l \neq i,j} (1.0 - \lambda(t))^2 \lambda(t) [[Q_{-i}][I - \lambda^2(t) Q_{-i,-j}]^{-1}]_{jl} \\
&\qquad\qquad [[Q_{-i,-j}][I - \lambda(t) Q_{-i,-j}]^{-1}]_{lj} \\
&\qquad\qquad (S_{l,j}(t) + S_{j,i}(t) - S_{j,j}(t) - S_{l,i}(t)) \\
&+ \sum_{l \neq i,j} \sum_{k \neq i} (1.0 - \lambda(t))^2 \lambda(t) [[Q_{-i}][I - \lambda^2(t) Q_{-i,-j}]^{-1}]_{jl} \lambda(t) \\
&\qquad [[Q_{-i}][I - \lambda(t) Q_{-i}]^{-1}]_{jk} [[Q_{-i,-j}][I - \lambda(t) Q_{-i,-j}]^{-1}]_{lj} \\
&\qquad\qquad (S_{k,l}(t) + S_{i,j}(t) - S_{i,l}(t) - S_{k,j}(t)) \\
&+ \sum_{l \neq i,j} \sum_{k \neq i} \Big((1.0 - \lambda(t)) \lambda^2(t) [[Q_{-i}][I - \lambda^2(t) Q_{-i,-j}]^{-1}]_{jl} \\
&\qquad [[Q_{-i,-j}][I - \lambda(t) Q_{-i,-j}]^{-1}]_{lj} \lambda(t) [[Q_{-i}][I - \lambda(t) Q_{-i}]^{-1}]_{jk} \\
&\qquad\qquad \Big(\mathbf{r1}_k (m_l - m_j) + q_k (S_{ij} - S_{il}) \Big) \\
&+ \sum_{l \neq i,j} (1.0 - \lambda(t)) [[Q_{-i}][I - \lambda^2(t) Q_{-i,-j}]^{-1}]_{jl} \\
&\qquad \lambda(t) [[Q_{-i,-j}][I - \lambda(t) Q_{-i,-j}]^{-1}]_{lj} \\
&\qquad\qquad \Big(\mathbf{r1}_j (m_l - m_j) + q_j (S_{ij} - S_{il}) \Big) \\
&+(1.0 - \lambda(t)) [[Q_{-i}][I - \lambda(t) Q_{-i,-j}]^{-1}]_{jj} (M_{jj} - M_{ij}), \qquad \text{(A.21)}
\end{aligned}
$$

where

$$
C_{ij}(t) \;=\; \frac{E_{ij}(t)}{1.0 - \lambda(t)[[Q_{-i}][I - \lambda(t) Q_{-i,-j}]^{-1}]_{jj}},
$$

and

$$
\begin{aligned}
F_{ij}(t) \;=\; & \sum_{l \neq i,j} (1.0 - \lambda(t)) [[Q_{-i}][I - \lambda(t) Q_{-i,-j}]^{-1}]_{jl} \\
&\qquad [[Q_{-i,-j}][I - Q_{-i,-j}]^{-1}]_{lj} [[Q_{-i}][I - Q_{-i}]^{-1}]_{ji} (M_{li} - M_{ij}) \\
&+ \sum_{l \neq i,j} (1.0 - \lambda(t)) [[Q_{-i}][I - \lambda(t) Q_{-i,-j}]^{-1}]_{jl} \\
&\qquad\qquad [[Q_{-i,-j}][I - Q_{-i,-j}]^{-1}]_{li} (M_{li} - M_{ji}) \\
&+(1.0 - \lambda(t)) [[Q_{-i}][I - \lambda(t) Q_{-i,-j}]^{-1}]_{ji} (M_{ii} - M_{ji})
\end{aligned}
$$

$$
\begin{aligned}
H_{ij}(t) \;=\; & \Big(F_{ij}(t) + \lambda(t) [[Q_{-i}][I - \lambda(t) Q_{-i,-j}]^{-1}]_{ji} C_{ji}(t) \Big) \\
&\qquad\qquad (1.0 - [[Q_{-j}][I - Q_{-j,-i}]^{-1}]_{ii}) \\
&+[[Q_{-i}][I - Q_{-i,-j}]^{-1}]_{ji} \\
&\qquad\qquad \Big(F_{ji}(t) + \lambda(t) [[Q_{-j}][I - \lambda(t) Q_{-j,-i}]^{-1}]_{ij} C_{ij}(t) \Big)
\end{aligned}
$$

$$
\begin{aligned}
X_{ij}(t) &= (1 - [[Q_{-i}][I - Q_{-i,-j}]^{-1}]_{jj})(1 - [[Q_{-j}][I - Q_{-j,-i}]^{-1}]_{ii}) \\
&\quad -[[Q_{-i}][I - Q_{-i,-j}]^{-1}]_{ji}[[Q_{-j}][I - Q_{-j,-i}]^{-1}]_{ij} \\
G_{ij}(t) &= \sum_{k \neq i,j} (1.0 - \lambda(t))[[Q_{-j}][I - \lambda(t)Q_{-j,-i}]^{-1}]_{ik} \\
&\qquad\qquad [[Q_{-j}][I - Q_{-j}]^{-1}]_{kj}(M_{kj} - M_{ij}) \\
&\quad +(1.0 - \lambda(t))[[Q_{-j}][I - \lambda(t)Q_{-j,-i}]^{-1}]_{ij}(M_{jj} - M_{ij}) \\
&\quad +\lambda(t)[[Q_{-j}][I - \lambda(t)Q_{-j,-i}]^{-1}]_{ij}C_{ij}(t) \\
&\quad +[[Q_{-j}][I - Q_{-j,-i}]^{-1}]_{ij}\frac{H_{ij}(t)}{X_{ij}(t)}.
\end{aligned}
$$

Finally,

$$
\begin{aligned}
\Upsilon^R_{ij}(t) &= (D_{-i,-j})_i \frac{G_{ij}}{1 - [[Q_{-j}][I - Q_{-j,-i}]^{-1}]_{ii}} \\
&\quad +(D_{-j,-i})_j \frac{G_{ji}}{1 - [[Q_{-i}][I - Q_{-i,-j}]^{-1}]_{jj}}.
\end{aligned}
$$

Notes

1. See Saul & Singh (1996) for learning curve bounds for an interesting Markov decision process that are derived using techniques from statistical mechanics.
2. There are other criteria for comparing algorithms, e.g., large deviation rates (Bucklew, 1990), but they are hard to compute for the TD algorithms, and in any case MSE is often reported.
3. Note that limiting the step-size to be a function of trial number alone prohibits $\alpha_i(t) = \frac{1}{\sum_{\tau=1}^{t} K_i(\tau)}$ or $\alpha_i(t) = \frac{1}{\sum_{\tau=1}^{t} \kappa_i(\tau)}$, as would be used in conventional first-visit MC and every-visit MC respectively. For these state dependent choices of α, Singh & Sutton (1996) showed that first-visit MC is unbiased while every-visit MC is biased but consistent, and that the variance of every-visit MC starts off less than or equal to the variance of first-visit MC but eventually becomes higher.
4. For convergence of accumulate TD, see Dayan & Sejnowski (1994), Jaakkola et al. (1994), Tsitsiklis (1994), Barnard (1993); and for convergence of replace TD, first-visit MC, and every-visit MC, see Singh & Sutton (1996). First TD converges appropriately because although its estimator uses a λ-weighted sum of multi-step returns that is different from replace TD and accumulate TD, its estimator remains a contraction in expected value, and therefore Jaakkola et al.'s (1994) convergence proof applies.
5. There are "importance sampling" methods for dealing with "difficult" distributions (see e.g., Bucklew, 1990), but it is not clear how they could be applied here.

References

Barnard, E. (1993). Temporal-difference methods and Markov models. *IEEE Transactions on Systems, Man, and Cybernetics*, *23*(2), 357–365.

Barto, A. G. and Duff, M. (1994). Monte Carlo matrix inversion and reinforcement learning. In *Advances in Neural Information Processing Systems 6*, pages 687–694, San Mateo, CA. Morgan Kaufmann.

Barto, A. G., Sutton, R. S., and Anderson, C. W. (1983). Neuronlike elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, *13*, 835–846.

Bucklew, J. A. (1990). *Large Deviation Techniques in Decision, Simulation and Estimation.* New York: Wiley-Interscience.

Dayan, P. (1992). The convergence of TD(λ) for general λ. *Machine Learning*, *8*(3/4), 341–362.

Dayan, P. and Sejnowski, T. (1994). TD(λ) converges with probability 1. *Machine Learning*, *14*, 295–301.

Haussler, D., Kearns, M., Seung, H. S., and Tishby, N. (1994). Rigorous learning curve bounds from statistical mechanics. In *Proceedings of the 7th Annual ACM Workshop on Computational Learning Theory*, pages 76–87, San Mateo, CA. Morgan Kauffman.

Jaakkola, T., Jordan, M. I., and Singh, S. (1994). On the convergence of stochastic iterative dynamic programming algorithms. *Neural Computation*, *6*(6), 1185–1201.

Saul, L. K. and Singh, S. (1996). Learning curves bounds for Markov decision processes with undiscounted rewards. In *Proceedings of COLT*.

Singh, S. and Sutton, R. S. (1996). Reinforcement learning with replacing eligibility traces. *Machine Learning*, Vol. 22, 123–158.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, *3*, 9–44.

Tsitsiklis, J. (1994). Asynchronous stochastic approximation and Q-learning. *Machine Learning*, *16*(3), 185–202.

Wasow, W. R. (1952). A note on the inversion of matrices by random walks. *Math. Tables Other Aids Comput.*, *6*, 78–81.

Watkins, C. J. C. H. (1989). *Learning from Delayed Rewards.* Ph.D Thesis, Cambridge Univ., Cambridge, England.

Widrow, B. and Stearns, S. D. (1985). *Adaptive Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall.