NOTE Communicated by Peter Dayan

A Counterexample to Temporal Differences Learning

Dimitri P. Bertsekas
Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 USA

Sutton's TD(λ) method aims to provide a representation of the cost function in an absorbing Markov chain with transition costs. A simple example is given where the representation obtained depends on λ. For $\lambda = 1$ the representation is optimal with respect to a least-squares error criterion, but as λ decreases toward 0 the representation becomes progressively worse and, in some cases, very poor. The example suggests a need to understand better the circumstances under which TD(0) and Q-learning obtain satisfactory neural network-based compact representations of the cost function. A variation of TD(0) is also given, which performs better on the example.

1 Introduction

We consider a Markov chain with states $0, 1, 2, \ldots, n$. The transition from state i to state j has probability p_{ij}, and cost $g(i,j)$. We assume that state 0 is cost-free and absorbing, and that it is eventually reached from every other state with probability one. In other words, $p_{00} = 1$ and $g(0,0) = 0$, and from every state i, there is a path of positive probability transitions that leads to 0. For each initial state i we want to estimate the expected total cost $J(i)$ up to reaching state 0.

We consider approximations within a class of differentiable functions $\tilde{J}(i,w)$ parameterized by a vector w. For example, $\tilde{J}(i,w)$ may be the output of a neural network when the input is i and the vector of weights is w. Sutton's TD(λ) method (Sutton 1988) is a gradient-like algorithm for obtaining a suitable vector w after observing a large number of simulated trajectories of the Markov chain. The method has attracted considerable attention, and has been used successfully in a more general setting by Tesauro (1992) for the training of a neural network to play backgammon. See Barto *et al.* (1994) for a nice and comprehensive survey of related issues.

For $\lambda \in [0,1]$, TD(λ) performs an infinite number of simulation runs, each ending at the absorbing state 0. Within the total number of runs, each state is encountered an infinite number of times. If $(i_1, i_2, \ldots, i_N, 0)$ is the typical trajectory, a positive stepsize γ is selected, and the vector

w is modified at the end of the kth transition by an increment that is proportional to γ and to the temporal difference d_k given by

$$d_k = g(i_k, i_{k+1}) + \tilde{J}(i_{k+1}, w) - \tilde{J}(i_k, w), \qquad k = 1, \ldots, N \tag{1.1}$$

where $i_{N+1} = 0$. The increment also involves the preceding gradients with respect to w, $\nabla\tilde{J}(i_m, w)$, $m = 1, \ldots, k$, which are evaluated at the vector w prevailing at the beginning of the simulation run. (An alternative possibility for which the analysis of this paper also holds is to evaluate these gradients at the current value of w.) The method is as follows:

Following the state transition (i_1, i_2), set

$$w := w + \gamma d_1 \nabla\tilde{J}(i_1, w) \tag{1.2}$$

Following the state transition (i_2, i_3), set

$$w := w + \gamma d_2 \left[\lambda\nabla\tilde{J}(i_1, w) + \nabla\tilde{J}(i_2, w)\right] \tag{1.3}$$

Following the state transition $(i_N, 0)$, set

$$w := w + \gamma d_N \left[\lambda^{N-1}\nabla\tilde{J}(i_1, w) + \lambda^{N-2}\nabla\tilde{J}(i_2, w) + \cdots + \nabla\tilde{J}(i_N, w)\right]$$
$$\cdots$$

By adding equations 1.2–1.4 for $\lambda = 1$, and by using the temporal differences formula 1.1, we see that the TD(1) iteration corresponding to a complete trajectory can be written as

$$w := w + \gamma \sum_{k=1}^{N} \left[\sum_{m=k}^{N} g(i_m, i_{m+1}) - \tilde{J}(i_k, w)\right] \nabla\tilde{J}(i_k, w)$$

so it is a gradient iteration for minimizing the sum of squares

$$\sum_{k=1}^{N} \left[\sum_{m=k}^{N} g(i_m, i_{m+1}) - \tilde{J}(i_k, w)\right]^2$$

It follows, as originally discussed by Sutton (1988), that TD(1) can be viewed as a form of incremental gradient or backpropagation method for minimizing over w the sum of the squared differences of the sample costs of the states i visited by the simulation and the estimates $\tilde{J}(i, w)$. This method has satisfactory convergence behavior, and is supported by classical results on stochastic approximation and stochastic gradient methods (see, e.g., Poljak and Tsypkin 1973; Kushner and Clark 1978; Poljak 1987; Bertsekas and Tsitsiklis 1989), and by more recent analyses on deterministic incremental gradient methods by Luo (1991), Luo and Tseng (1993), and Mangasarian and Solodov (1993). Thus TD(1) will typically tend to yield a value of w that minimizes a weighted sum of the squared errors

$$|c(i) - \tilde{J}(i, w)|^2$$

where $c(i)$ is the average sample cost corresponding to state i, and the weights of different states are determined by the relative frequencies these states are visited during the simulation. An alternative view that leads to similar conclusions is to consider TD(1) as a stochastic gradient method for minimizing an expected value of the square of the error $J(i) - \tilde{J}(i,w)$.

On the other hand for $\lambda < 1$, the convergence behavior of TD(λ) is unclear, unless w contains enough parameters to make possible an exact representation of $J(i)$ by $\tilde{J}(i,w)$ for all states i (a lookup table representation), as shown in various forms by Sutton (1988), Dayan (1992), Tsitsiklis (1993), and Jaakkola *et al.* (1993). Actually, Sutton's and Dayan's convergence results apply to the slighly more general case of linear representations, under a restrictive linear independence condition on the set of observation vectors. Basically, TD(λ) can be viewed as a form of incremental gradient method where there are some error terms in the gradient direction. These error terms depend on w as well as λ, and they typically do not diminish when w is equal to the value where TD(1) converges, unless $\lambda = 1$ or a lookup table representation is used. Thus, in general, the limit obtained by TD(λ) depends on λ, as has also been shown by Dayan (1992). Nonetheless, there are accounts of good practical performance of TD(λ), even with λ substantially less than 1. For example, Tesauro (1992) reports that his backgammon program performs better when trained with small than with high values of λ.

2 An Example

In the following example we use a linear approximation of the form

$$\tilde{J}(i,w) = iw$$

and we find that as λ is reduced from the value 1, TD(λ) converges to an increasingly poor value $\hat{w}(\lambda)$. For a deliberate choice of the problem data, we obtain

$$\hat{w}(0) \approx -\hat{w}(1)$$

that is, a reversal of sign of $\tilde{J}(i,w)$ (see Fig. 2).

In our example the state transitions and associated costs are deterministic. In particular, from state i we move to state $i-1$ with a given cost g_i. Let all simulation runs start at state n and end at 0 after visiting all the states $n-1, n-2, \ldots, 1$ in succession. The temporal difference associated with the transition from i to $i-1$ is

$$g_i + \tilde{J}(i-1,w) - \tilde{J}(i,w) = g_i - w$$

and the corresponding gradient is

$$\nabla \tilde{J}(i,w) = i$$

The iteration of TD(λ) corresponding to a complete trajectory is given by

$$w := w + \gamma \sum_{k=1}^{n} (g_k - w) \left[\lambda^{n-k} n + \lambda^{n-k-1}(n-1) + \cdots + k \right] \tag{2.1}$$

and is linear in w.

Suppose that the stepsize γ is either constant and satisfies

$$0 < \gamma < \left\{ \sum_{k=1}^{n} \left[\lambda^{n-k} n + \lambda^{n-k-1}(n-1) + \cdots + k \right] \right\}^{-1}$$

(in which case the iteration 2.1 is contracting), or else γ is diminishing at a rate that is inversely proportional to the number of simulation runs performed thus far. Then the TD(λ) iteration 2.1 converges to the scalar $\hat{w}(\lambda)$ for which the increment in the right-hand side of equation 2.1 is zero, that is,

$$\sum_{k=1}^{n} [g_k - \hat{w}(\lambda)] \left[\lambda^{n-k} n + \lambda^{n-k-1}(n-1) + \cdots + k \right] = 0$$

In particular, we have

$$\hat{w}(1) = \frac{n(g_1 + \cdots + g_n) + (n-1)(g_1 + \cdots + g_{n-1}) + \cdots + g_1}{n^2 + (n-1)^2 + \cdots + 1} \tag{2.2}$$

$$\hat{w}(0) = \frac{n g_n + (n-1) g_{n-1} + \cdots + g_1}{n + (n-1) + \cdots + 1} \tag{2.3}$$

It can be seen that $\hat{w}(1)$ minimizes over w the sum of squared errors

$$\sum_{i=1}^{n} \left| J(i) - \tilde{J}(i, w) \right|^2 \tag{2.4}$$

where

$$J(i) = g_1 + \cdots + g_i, \qquad \tilde{J}(i, w) = iw, \qquad \forall\, i = 1, \ldots, n$$

Indeed the optimality condition for minimization of the function 2.4 over w is

$$\sum_{i=1}^{n} i(g_1 + \cdots + g_i - iw) = 0$$

which when solved for w gives a solution equal to $\hat{w}(1)$ as given by equation 2.2.

Figures 1 and 2 show the form of the cost function $J(i)$, and the representations $\tilde{J}[i, \hat{w}(1)]$ and $\tilde{J}[i, \hat{w}(0)]$ provided by TD(1) and TD(0), respectively, for $n = 50$ and for the following two cases:

1. $g_1 = 1, \qquad g_i = 0, \quad \forall\, i \neq 1$
2. $g_n = -(n-1), \qquad g_i = 1, \quad \forall\, i \neq n$

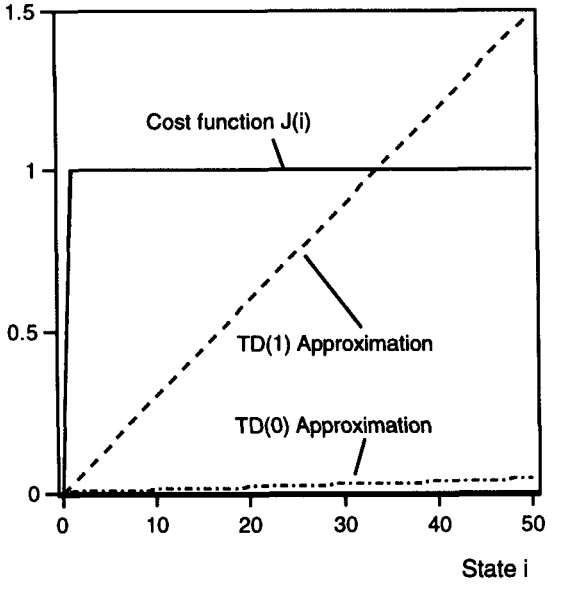

Figure 1: Form of the cost function $J(i)$, and the linear representations $\tilde{J}[i, \hat{w}(1)]$, and $\tilde{J}[i, \hat{w}(0)]$ provided by TD(1) and TD(0), respectively, for the case $g_1 = 1$, $g_i = 0, \forall\, i \neq 1$.

It can be seen that TD(0) can yield a very poor approximation to the cost function.

The above example can be generalized with similar results. For instance, the cost of transition from i to $i-1$ may be random, in which case the costs g_i must be replaced by their expected values in equations 2.2 and 2.3. The trajectories need not all start at state n. The results are qualitatively similar if the successor state of state i is randomly chosen. Also, similar behavior can be observed in a variety of stochastic examples that can be constructed with our deterministic example as a "building block."

The example indicates that for $\lambda < 1$, TD(λ) is in need of further justification for the case of a compact cost function representation. The example also relates to one of Watkins' Q-learning methods (Watkins 1989). These methods have the advantage that they apply to discounted Markovian decision problems and stochastic shortest path problems (as defined in Bertsekas and Tsitsiklis 1989), where there are multiple actions available at each state and the objective is not just to obtain the optimal cost, but also to find an optimal action at each state. Strong convergence results have been recently shown by Tsitsiklis (1993) for the

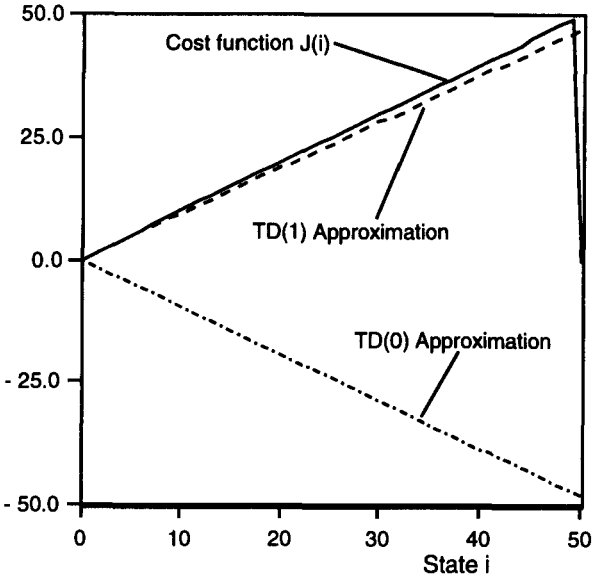

Figure 2: Form of the cost function $J(i)$, and the linear representations $\tilde{J}[i, \hat{w}(1)]$, and $\tilde{J}[i, \hat{w}(0)]$ provided by TD(1) and TD(0), respectively, for the case $g_n = -(n-1)$, $g_i = 1$, $\forall\, i \neq n$.

most commonly used Q-learning method in the case of a lookup table representation. TD(0) can be viewed as a special case of this Q-learning method for the situation where there is only one action available at each state, so our conclusions also apply to the corresponding neural network versions.

3 A Partial Remedy

In view of the preceding example, it is interesting to ask whether there is a modified version of TD(0) that yields the exact cost values in the case of a lookup table representation and approximates the cost values better when compact representations are used.

For the case of a lookup table representation, we know that TD(0) can be viewed as a Robbins–Monro method for solving the system of equations

$$\sum_{i=1}^{n} p_{ij}\left[g(i,j) + \tilde{J}(j,w)\right] - \tilde{J}(i,w) = 0, \qquad i = 1, \ldots, n \tag{3.1}$$

that is, for finding a w for which the expected value of the temporal difference vanishes at each state i. For the case of a compact representation, it is thus reasonable to consider a weighted least-squares problem that aims at making the size of the expected temporal differences small in an aggregate sense, that is, a problem of the form

$$\min_w \sum_{i=1}^n q_i |E_j\{d(i,j) \mid i\}|^2 = \sum_{i=1}^n q_i \left| \sum_{j=1}^n p_{ij} \left[g(i,j) + \tilde{J}(j,w) \right] - \tilde{J}(i,w) \right|^2 \quad (3.2)$$

where

$$d(i,j) = g(i,j) + \tilde{J}(j,w) - \tilde{J}(i,w)$$

denotes the temporal difference associated with the transition from i to j, $E_j\{\cdot \mid i\}$ denotes conditional expected value over j given i, and q_i is a nonnegative weight for each state i.

A simulation-based gradient method for solving such a problem is to update w following a transition from i_k by the iteration

$$\begin{aligned} w &:= w + \gamma \left[\sum_{j=1}^n p_{i_k j} d(i_k, j) \right] \left[\nabla \tilde{J}(i_k, w) - \sum_{j=1}^n p_{i_k j} \nabla \tilde{J}(j, w) \right] \\ &= w + \gamma E_j\{d(i_k, j) \mid i_k\} \left(\nabla \tilde{J}(i_k, w) - E_j \left\{ \nabla \tilde{J}(j, w) \mid i_k \right\} \right) \end{aligned} \quad (3.3)$$

The relative frequencies of visits to different states determine the relative weights in the corresponding least-squares problem (3.2). Note that the *expected temporal difference*

$$E_j\{d(i_k, j) \mid i_k\} = \sum_{j=1}^n p_{i_k j} d(i_k, j)$$

at i_k and the *expected gradient*

$$E_j\{\nabla \tilde{J}(j,w) \mid i_k\} = \sum_{j=1}^n p_{i_k j} \nabla \tilde{J}(j,w)$$

over the successor states j appear in the right-hand side of this iteration. Thus the computational requirements per iteration are increased over TD(λ), unless the system is deterministic. The method (3.3) is apparently new, although an iteration similar to 3.3 and its sampled version given below (cf. equation 3.6) have been independently developed by Baird and are briefly described in Baird (1993) and Harmon *et al.* (1994) (this was pointed out by one of the reviewers).

For the deterministic example of Section 2, the iteration 3.3 takes the form

$$w := w + \gamma(g_k - w)\,[k - (k-1)] = w + \gamma(g_k - w)$$

so the iteration corresponding to a full trajectory $(n, n-1, \ldots, 1, 0)$ is

$$w := w + \gamma \sum_{k=1}^{n} (g_k - w) \tag{3.4}$$

When γ is smaller than $1/n$, this iteration converges to

$$\hat{w} = \frac{\sum_{k=1}^{n} g_k}{n} \tag{3.5}$$

This corresponds to a linear approximation, which is exact for state n, that is, $J(n) = \tilde{J}(n, \hat{w})$, regardless of the costs of the other states. In particular, for the example of Figure 1, we obtain in the limit $\hat{w} = 1/n$, while for the example of Figure 2, we obtain $\hat{w} = 0$. The corresponding approximations $\tilde{J}(i, \hat{w}) = i\hat{w}$ are not as good as those obtained by TD(1), but they are much better than those obtained by TD(0). While it is unclear whether such a conclusion can be reached in a more general setting, in the author's limited experimentation with some stochastic problems, iteration 3.3 has produced substantially better compact cost representations than TD(0).

There is a simpler version of iteration 3.3 that does not require averaging over the successor states j. In this version, the two expected values in iteration 3.3 are replaced by two independent single sample values. In particular, w is updated by

$$w := w + \gamma d(i_k, i_{k+1}) \left[\nabla \tilde{J}(i_k, w) - \nabla \tilde{J}(i'_{k+1}, w) \right] \tag{3.6}$$

where i_{k+1} and i'_{k+1} correspond to two independent transitions starting from i_k. It can be seen that this iteration yields in the limit values of w that solve the least-squares problem (3.2). It is necessary to use two independently generated states i_{k+1} and i'_{k+1} in order that the expected value of the product $d(i_k, i_{k+1}) \left[\nabla \tilde{J}(i_k, w) - \nabla \tilde{J}(i'_{k+1}, w) \right]$ given i_k is equal to the term $E_j\{d(i_k, j) \mid i_k\} \left(\nabla \tilde{J}(i_k, w) - E_j \left\{ \nabla \tilde{J}(j, w) \mid i_k \right\} \right)$ appearing in the right-hand side of equation 3.3.

The variant of iteration 3.6 where a single sample ($i_{k+1} = i'_{k+1}$) is used, that is,

$$w := w + \gamma d(i_k, i_{k+1}) \left[\nabla \tilde{J}(i_k, w) - \nabla \tilde{J}(i_{k+1}, w) \right] \tag{3.7}$$

has been discussed by Dayan (1992). It aims at solving the problem

$$\min_{w} \sum_{i=1}^{n} q_i E_j \left\{ |d(i, j)|^2 \mid i \right\} = \sum_{i=1}^{n} q_i \sum_{j=1}^{n} p_{ij} \left| g(i, j) + \tilde{J}(j, w) - \tilde{J}(i, w) \right|^2 \tag{3.8}$$

where q_i are the nonnegative weights also appearing in equation 3.2, which are determined by the relative frequencies of the visits to different states during the simulation. This problem involves a weighted sum of second moments of the temporal differences, which is not as desirable an objective as the weighted sum of the squares of the means of the temporal differences, which is minimized by iteration 3.3. In particular, in the case

of a lookup table representation, iteration 3.3 yields the exact cost values, while solving problem 3.8 can give other values that may also depend on the weights q_i. Thus it appears that iteration 3.7 is unsuitable for Markov chains that are not deterministic.

Acknowledgments

Research supported by the SBIR through a contract with Alphatech, Inc.

References

Barto, A. G., Bradtke, S. J., and Singh, S. P. 1994. Learning to act using real-time dynamic programming. *J. Artificial Intelligence*, in press.

Baird, L. C. 1993. *Advantage updating*. Tech. Rep. WL-TR-93-1146, Wright Lab., Wright-Patterson Air Force Base, OH.

Bertsekas, D. P., and Tsitsiklis, J. N. 1989. *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, Englewood Cliffs, NJ.

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

Harmon, M. E., Baird, L. C., and Klopf, A. H. 1994. Advantage updating applied to a differential game. *NIPS Conf.*, Denver, Colorado, submitted.

Jaakkola, T., Jordan, M. I., and Singh, S. P. 1993. *On the convergence of stochastic iterative dynamic programming algorithms*. MIT Computational Cognitive Science Tech. Rep. 9307.

Kushner, H. J., and Clark, D. S. 1978. *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Springer-Verlag, New York.

Luo, Z. Q. 1991. On the convergence of the LMS algorithm with adaptive learning rate for linear feedforward networks. *Neural Comp.* **3**, 226–245.

Luo, Z. Q., and Tseng, P. 1993. Analysis of an approximate gradient projection method with applications to the back propagation algorithm. Department of Electrical and Computer Engineering, McMaster University, Hamilton, Ontario and Department of Mathematics, University of Washington, Seattle.

Mangasarian, O. L., and Solodov, M. V. 1993. *Serial and parallel backpropagation convergence via nonmonotone perturbed minimization*. Computer Science Department, Computer Sciences Tech. Rep. No. 1149, University of Wisconsin-Madison, April 1993.

Poljak, B. T. 1987. *Introduction to Optimization*. Optimization Software Inc., New York.

Poljak, B. T., and Tsypkin, Y. Z. 1973. Pseudogradient adaptation and training algorithms. *Automation Remote Control* 45–68.

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

Tesauro, G. 1992. Practical issues in temporal difference learning. *Machine Learn.* **8**, 257–277.

Tsitsiklis, J. N. 1993. Asynchronous stochastic approximation and Q-learning. LIDS Report P-2172, MIT.
Watkins, C. J. C. H. 1989. Learning from delayed rewards. Ph.D. Thesis, Cambridge University, England.

Received April 21, 1994; accepted July 22, 1994.