Temporal Difference Learning

Introduction

- TD learning is an unsupervised technique in which the learning agent learns to predict the expected value of a variable occurring at the end of a sequence of states
- Reinforcement learning (RL) extends this technique by allowing the learned state-values to guide actions which subsequently change the environment state
- The name TD derives from its use of changes, or differences, in predictions over successive time steps to drive the learning process
- TD learning is most closely associated with R. S. Sutton, whose 1984 Ph.D. dissertation addressed TD learning and whose 1988 paper, in which the term Temporal Difference was first used, has become the definitive reference

Introduction

- TD learning is a combination of Monte Carlo ideas and dynamic programming ideas
    - Like Monte Carlo methods, TD methods can learn directly from raw experience without a model of the environment's dynamics
    - Like DP, TD methods update estimates based in part on other learned estimates, without waiting for a final outcome (bootstrapping)

TD Prediction

- Monte Carlo Method
  - Wait until the end of the episode, retrieve the accumulated rewards Gt and update V(st)
    - $V(st) \leftarrow V(st) + \alpha[Gt - V(st)]$
- TD(0)
  - Form a target and make a useful update using the observed reward rt and the estimate V(st)
    - $V(st) \leftarrow V(st) + \alpha[rt+1 + \gamma V(st+1) - V(st)]$

TD Prediction

$$
\begin{aligned}
V^{\pi}(s) &= E_{\pi}\{R_t | s_t = s\} && (6.3)\\
&= E_{\pi}\left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \;\middle|\; s_t = s \right\} \\
&= E_{\pi}\left\{ r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \;\middle|\; s_t = s \right\} \\
&= E_{\pi}\left\{ r_{t+1} + \gamma V^{\pi}(s_{t+1}) \;\middle|\; s_t = s \right\}. && (6.4)
\end{aligned}
$$

TD methods combine the sampling of Monte Carlo with the bootstrapping of DP

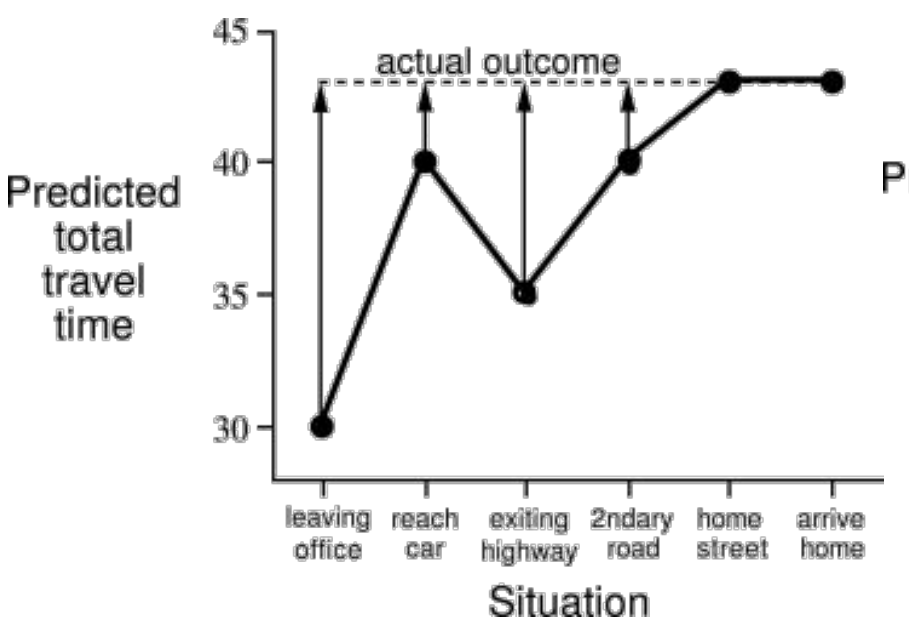
Procedural Form

Initialize $V(s)$ arbitrarily, π to the policy to be evaluated
Repeat (for each episode):
  Initialize s
  Repeat (for each step of episode):
    $a \leftarrow$ action given by π for s
    Take action a; observe reward, r, and next state, s'
    $V(s) \leftarrow V(s) + \alpha\left[r + \gamma V(s') - V(s)\right]$
    $s \leftarrow s'$
  until s is terminal

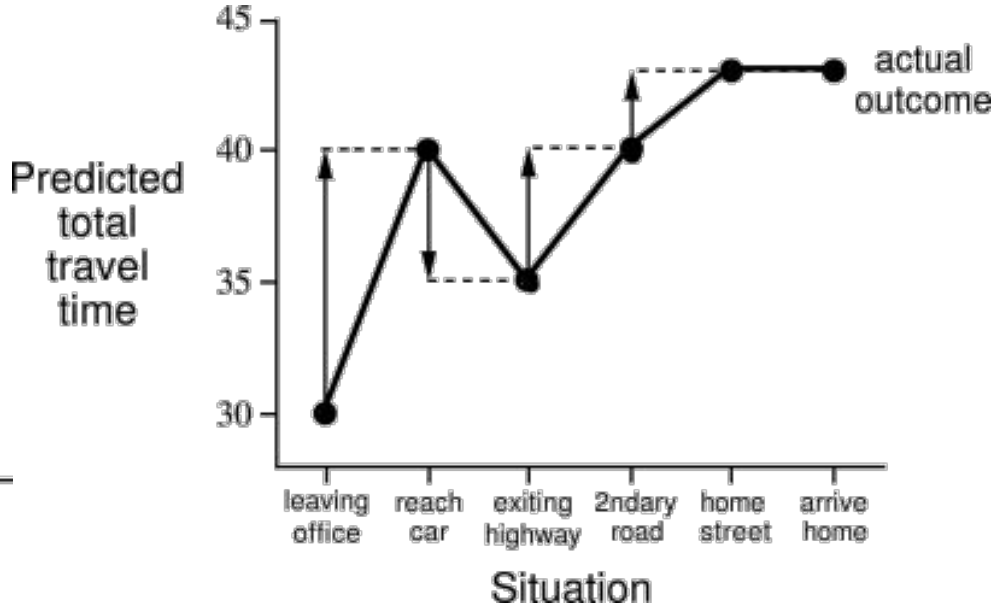
Example

	Elapsed Time	*Predicted*	*Predicted*
State	*(minutes)*	*Time to Go*	*Total Time*
leaving office, friday at 6	0	30	30
reach car, raining	5	35	40
exiting highway	20	15	35
2ndary road, behind truck	30	10	40
entering home street	40	3	43
arrive home	43	0	43

Changes Recommended



MC



TD

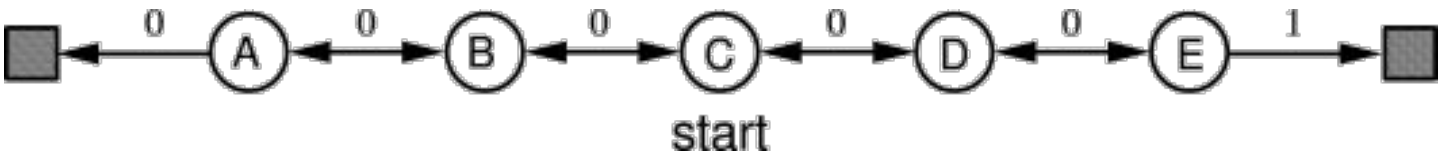
Advantages of TD Prediction Methods

- Obviously, TD methods have an advantage over DP methods in that they do not require a model of the environment, of its reward and next-state probability distributions
- They are naturally implemented in an on-line, fully incremental fashion
- With Monte Carlo methods one must wait until the end of an episode, because only then is the return known, whereas with TD methods one need wait only one time step

Advantages of TD Prediction Methods

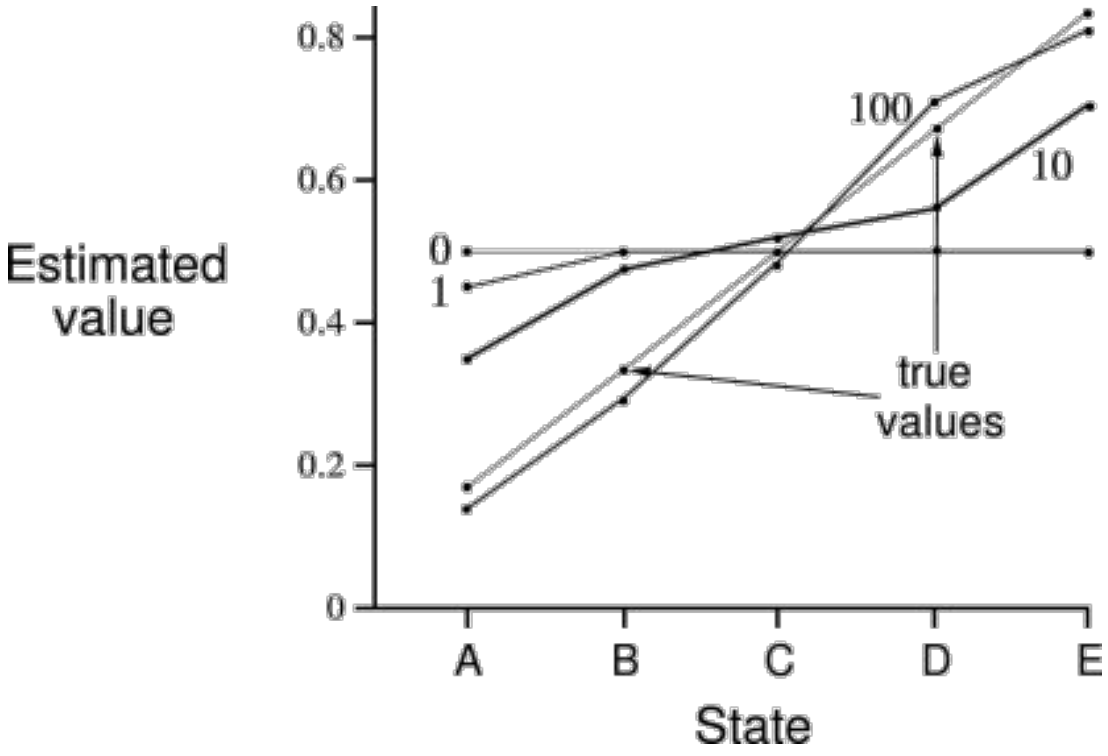
- Obviously, TD methods have an advantage over DP methods in that they do not require a model of the environment, of its reward and next-state probability distributions
- They are naturally implemented in an on-line, fully incremental fashion
- With Monte Carlo methods one must wait until the end of an episode, because only then is the return known, whereas with TD methods one need wait only one time step
- TD methods have usually been found to converge faster than constant-MC methods on stochastic tasks

Example

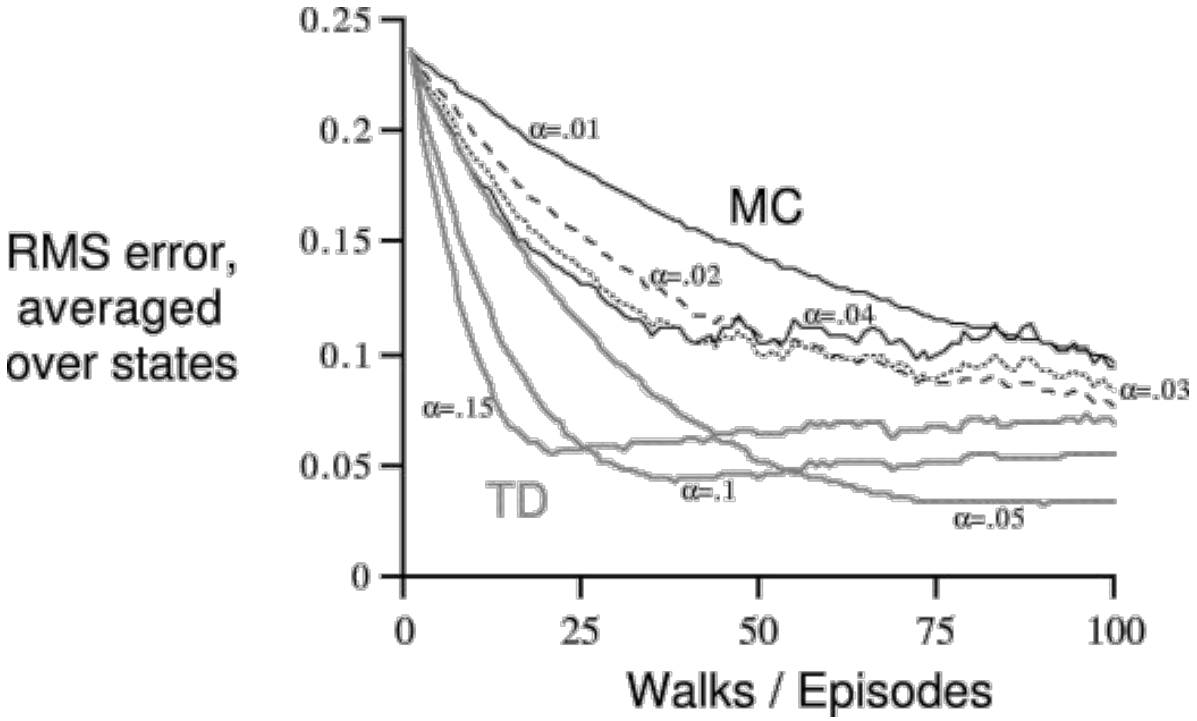


- All episodes start in the center state C and proceed either left or right by one state on each step, with equal probability
- Episodes terminate either on the extreme left or the extreme right

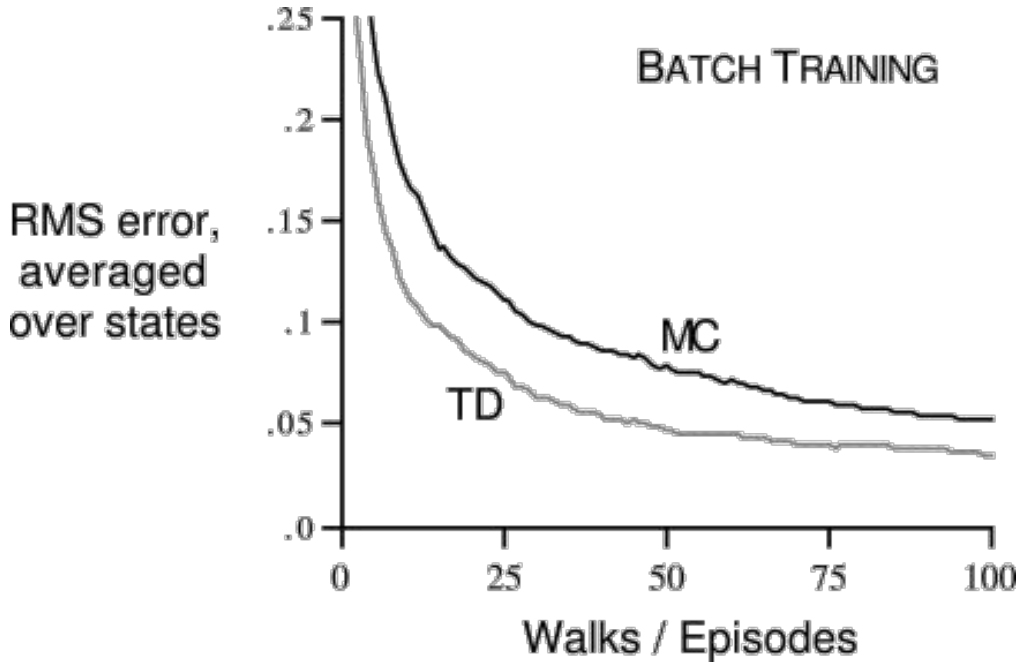
Example


Estimated value
0.8
0.6
0.4
0.2
0
100
10
0
1
true values
A
B
C
D
E
State

Example

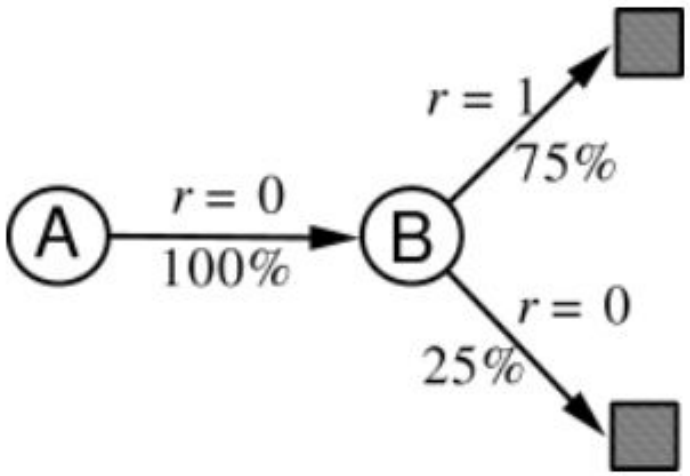

RMS error, averaged over states
0.25
0.2
0.15
0.1
0.05
0
0
25
50
75
100
Walks / Episodes
MC
TD
α=.01
α=.02
α=.04
α=.03
α=.15
α=.1
α=.05

Example



- Under batch updating, TD(0) converges deterministically to a single answer independent of the step-size parameter, alpha, as long it is chosen to be sufficiently small

Example



$A, 0, B, 0$
$B, 1$
$B, 1$
$B, 1$
$B, 1$
$B, 1$
$B, 1$
$B, 0$

What is the optimal value for the estimate V(A) given this data?

Optimality of TD(0)

- Batch Monte Carlo methods always find the estimates that minimize MSE on the training set, whereas batch TD(0) always finds the estimates that would be exactly correct for the maximum-likelihood model of the Markov process
- Batch TD(0) converges to the certainty-equivalence estimate
    - It is equivalent to assuming that the estimate of the underlying process was known with certainty rather than being approximated
- On tasks with large state spaces, TD methods may be the only feasible way of approximating the certainty-equivalence solution

References

- Richard S. Sutton and Andrew G. Barto (2017)
  Reinforcement Learning: An Introduction

Thank you