10703 Deep Reinforcement Learning

Tom Mitchell

Machine Learning Department

September 17, 2018

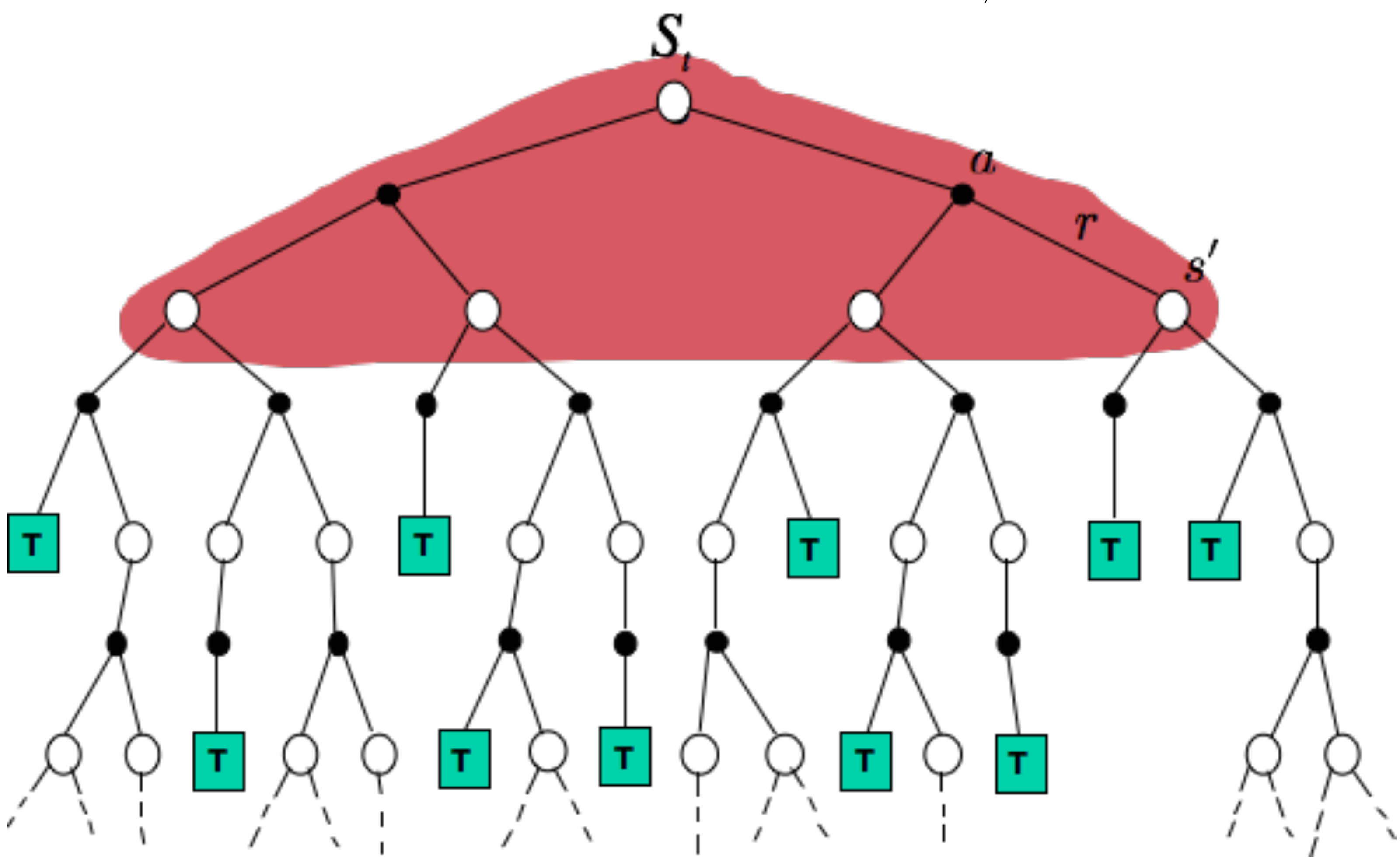
Temporal Difference Methods

Used Materials

• **Acknowledgement**: Much of the material and slides for this lecture were borrowed from Ruslan Salakhutdinov, who in turn borrowed much from Rich Sutton's class and David Silver's class on Reinforcement Learning.

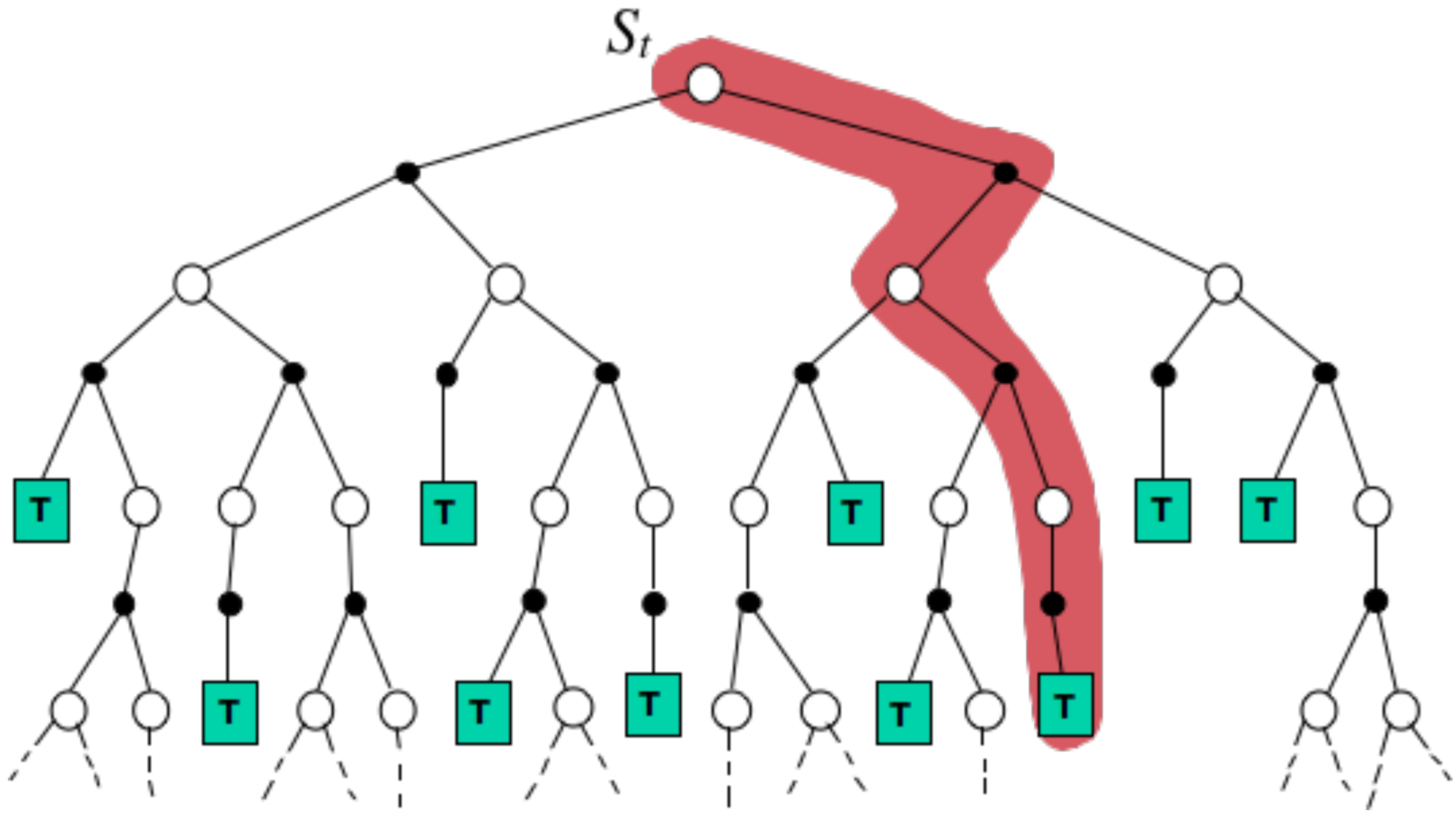
Dynamic Programming

$$V(S_t) \leftarrow E_\pi\Big[R_{t+1} + \gamma V(S_{t+1})\Big] = \sum_a \pi(a|S_t) \sum_{s',r} p(s',r|S_t,a)[r + \gamma V(s')]$$



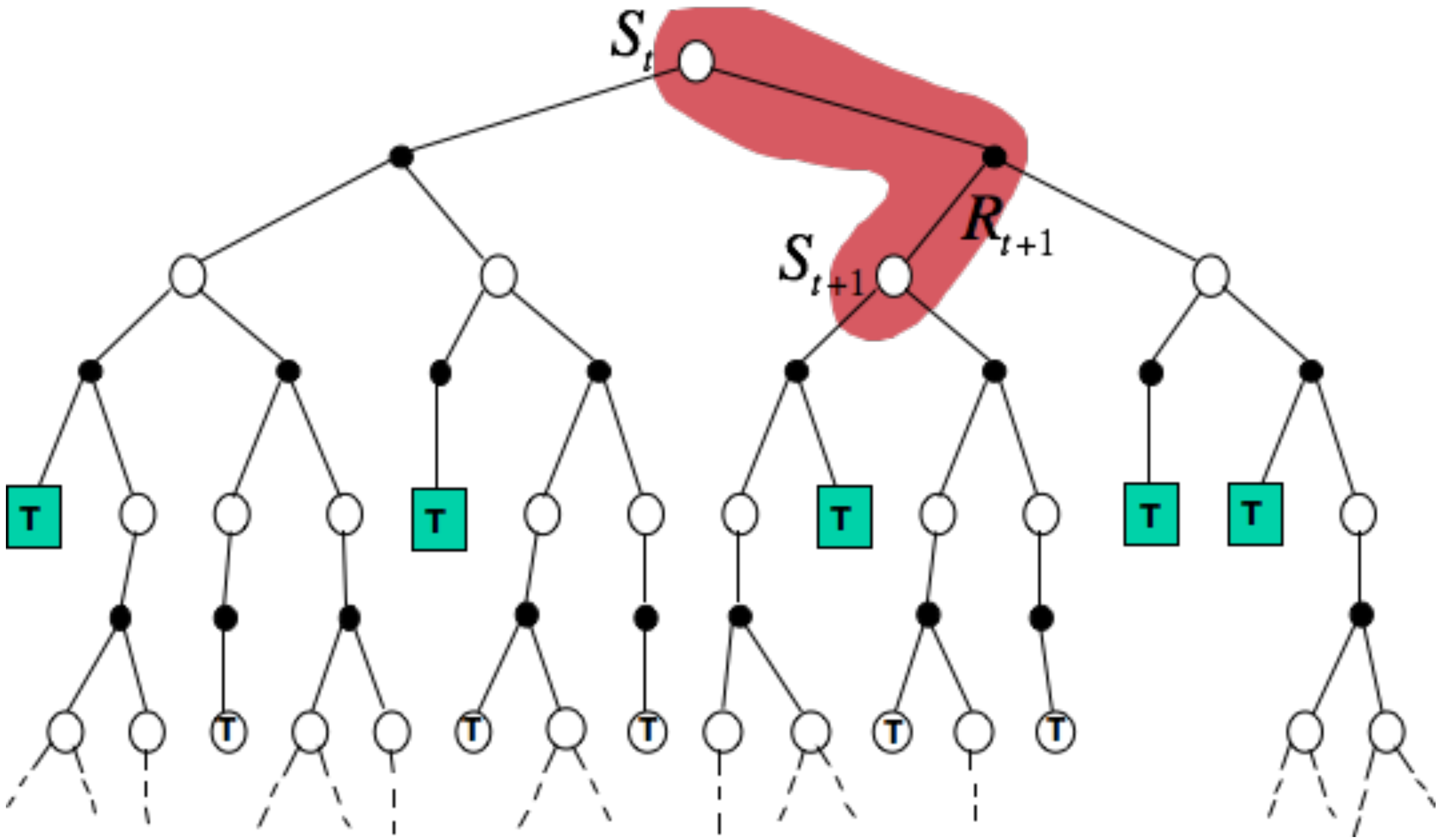
Monte Carlo

$$V(S_t) \leftarrow V(S_t) + \alpha\left(G_t - V(S_t)\right)$$



Simplest TD(0) Method

$$V(S_t) \leftarrow V(S_t) + \alpha \left(R_{t+1} + \gamma V(S_{t+1}) - V(S_t) \right)$$



TD Prediction

- Policy Evaluation (the prediction problem):
  - for a given policy π, compute the state-value function v_π
- Remember: Simple every-visit Monte Carlo method:

$$V(S_t) \leftarrow V(S_t) + \alpha \Big[G_t - V(S_t) \Big]$$

target: the actual return after time *t*

- The simplest Temporal-Difference method TD(0):

$$V(S_t) \leftarrow V(S_t) + \alpha \Big[\underbrace{R_{t+1} + \gamma V(S_{t+1})} - V(S_t) \Big]$$

target: an estimate of the return

TD(0) Prediction

- Policy Evaluation (the prediction problem):
  - for a given policy π, compute the state-value function v_π

- The simplest Temporal-Difference method TD(0):

$$V(S_t) \leftarrow V(S_t) + \alpha\Big[\underbrace{R_{t+1} + \gamma V(S_{t+1})} - V(S_t)\Big]$$

- Or estimate $Q(S_t,A_t)$ with TD(0)

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[\underbrace{R_{t+1} + \gamma \max_{a\in A} Q(S_{t+1}, a)}_{\textbf{target}\text{: an estimate of the return}} - Q(S_t, A_t)]$$

Q-Learning: Off-Policy TD(0) Control

- One-step Q-learning:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\Big[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)\Big]$$

Initialize $Q(s, a), \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
    Initialize S
    Repeat (for each step of episode):
        Choose A from S using policy derived from Q (e.g., ε-greedy)
        Take action A, observe R, S'
        $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_a Q(S', a) - Q(S, A)]$
        $S \leftarrow S'$;
    until S is terminal

- Converges, if every (s,a) pair is visited infinitely often

Many ways to blend sampled R_{t+k} and model-based estimates $Q(s_{t+k},a)$

Possible target values for training $Q(s_t,a_t)$:

$$[R_{t+1} + \gamma \max_{a \in A} Q(S_{t+1}, a)]$$

$$[R_{t+1} + \gamma R_{t+2} + \gamma^2 \max_{a \in A} Q(S_{t+2}, a)]$$

$$[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 \max_{a \in A} Q(S_{t+3}, a)]$$

$$\ldots$$

Many ways to blend sampled R_{t+k} and current estimates of $Q(s_{t+k},a)$

Possible target values for training $Q(s_t,a_t)$:

$$[R_{t+1} + \gamma \max_{a \in A} Q(S_{t+1}, a)]$$

$$[R_{t+1} + \gamma R_{t+2} + \gamma^2 \max_{a \in A} Q(S_{t+2}, a)]$$

$$[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 \max_{a \in A} Q(S_{t+3}, a)]$$

$$\ldots$$

TD(λ) – Blend all of these

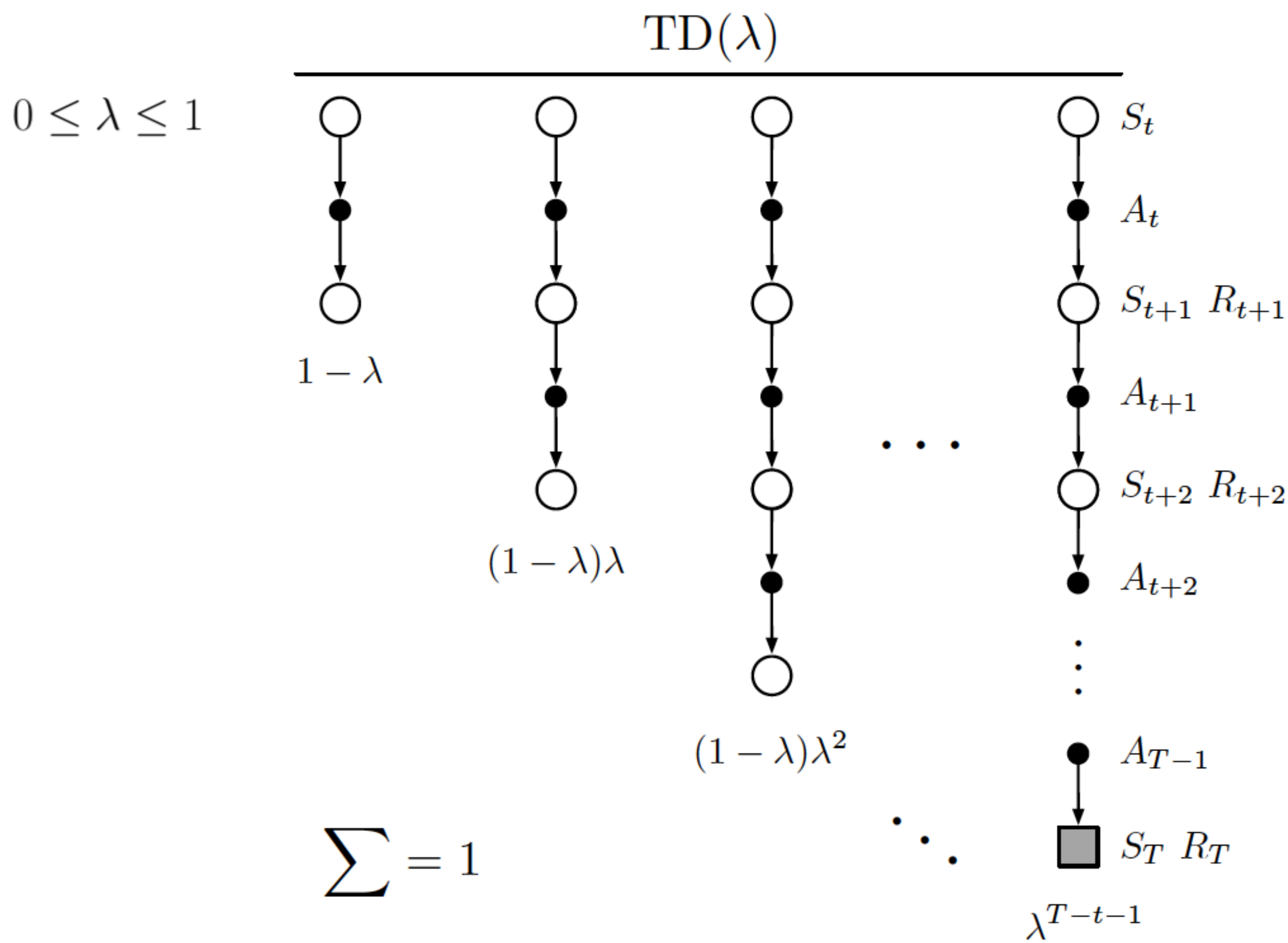


Figure 12.1: The backup digram for TD(λ). If $\lambda = 0$, then the overall update reduces to its first component, the one-step TD update, whereas if $\lambda = 1$, then the overall update reduces to its last component, the Monte Carlo update.

$$
\begin{aligned}
&(1-\lambda)\left[R_{t+1}+\gamma \max_{a\in A} Q(S_{t+1},a)\right] \\
&+(1-\lambda)\lambda\left[R_{t+1}+\gamma R_{t+2}+\gamma^2 \max_{a\in A} Q(S_{t+2},a)\right] \\
&+(1-\lambda)\lambda^2\left[R_{t+1}+\gamma R_{t+2}+\gamma^2 R_{t+3}+\gamma^3 \max_{a\in A} Q(S_{t+3},a)\right] \\
&+\dots
\end{aligned}
$$

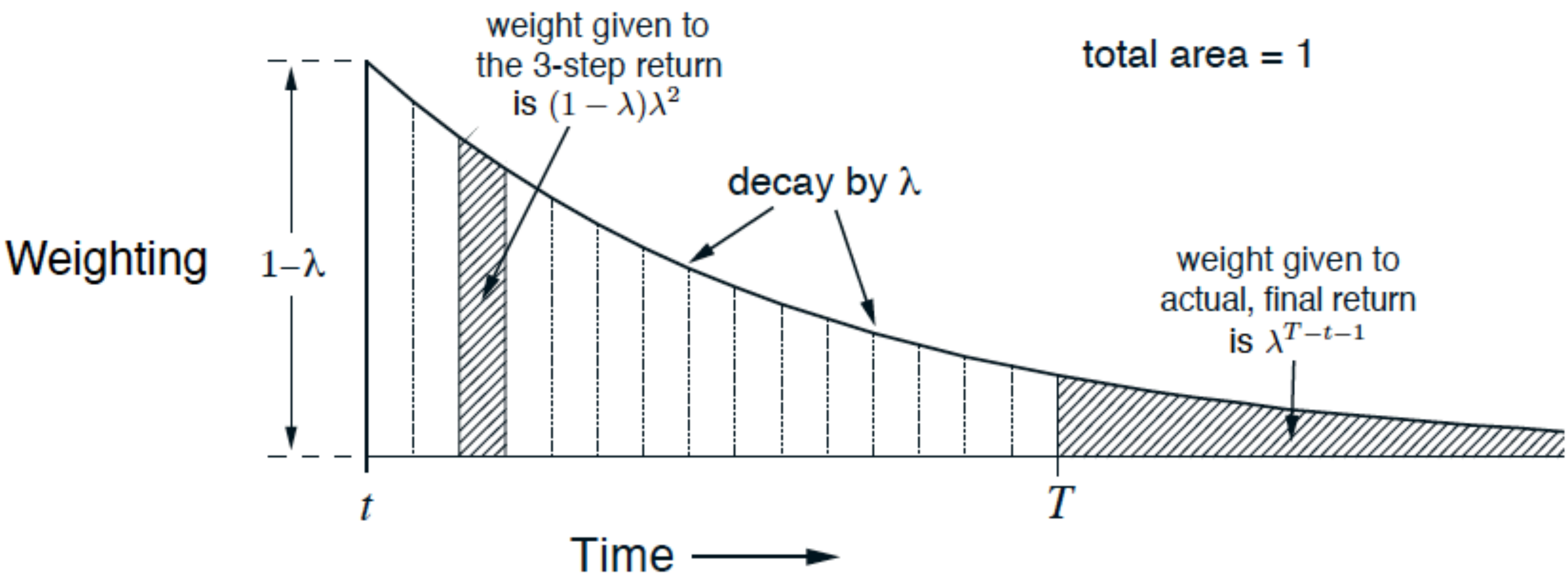


Figure 12.2: Weighting given in the λ-return to each of the n-step returns.

Bias-Variance Trade-Off

- MC has high variance, zero bias
  - Good convergence properties
  - Even with function approximation
  - Not very sensitive to initial value
  - Very simple to understand and use

- TD has low variance, some bias
  - Usually more efficient than MC
  - More sensitive to initial values of Q(s,a) and V(s)