ECE 414/517 Reinforcement Learning

LECTURE 12: MONTE CARLO METHODS

OCT. 10 2022

Monte Carlo Summary

1. MC has several advantages over DP:
    - Can learn directly from interaction with the environment
    - No need for full models
    - No need to learn about all states
    - Less harm by Markovian Violations (more details later).
2. One issue to watch out for: Maintaining sufficient exploration
    - Exploring starts, soft policies.
3. Introduced the distinction between on-policy and off-policy methods.
    - Why?
    - Why not?
4. No bootstrapping (as opposed to DP).

Monte Carlo (Exploring Starts)

Monte Carlo ES (Exploring Starts), for estimating $\pi \approx \pi_*$

Initialize:
  $\pi(s) \in \mathcal{A}(s)$ (arbitrarily), for all $s \in \mathcal{S}$
  $Q(s,a) \in \mathbb{R}$ (arbitrarily), for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$
  $Returns(s,a) \leftarrow$ empty list, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

Loop forever (for each episode):
  Choose $S_0 \in \mathcal{S}$ and $A_0 \in \mathcal{A}(S_0)$ such that all pairs have probability > 0
  Generate an episode from S_0, A_0, following π: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
  $G \leftarrow 0$
  Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
    $G \leftarrow \gamma G + R_{t+1}$
    Unless the pair S_t, A_t appears in $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:
      Append G to $Returns(S_t, A_t)$
      $Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$
      $\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$

On policy Monte Carlo

On-policy first-visit MC control (for ε-soft policies), estimates $\pi \approx \pi_*$

Algorithm parameter: small $\varepsilon > 0$

Initialize:

  $\pi \leftarrow$ an arbitrary ε-soft policy

  $Q(s,a) \in \mathbb{R}$ (arbitrarily), for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

  $Returns(s,a) \leftarrow$ empty list, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

Repeat forever (for each episode):

  Generate an episode following π: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$

  $G \leftarrow 0$

  Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:

    $G \leftarrow \gamma G + R_{t+1}$

    Unless the pair S_t, A_t appears in $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:

      Append G to $Returns(S_t, A_t)$

      $Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$

      $A^* \leftarrow \arg\max_a Q(S_t, a)$ (with ties broken arbitrarily)

      For all $a \in \mathcal{A}(S_t)$:

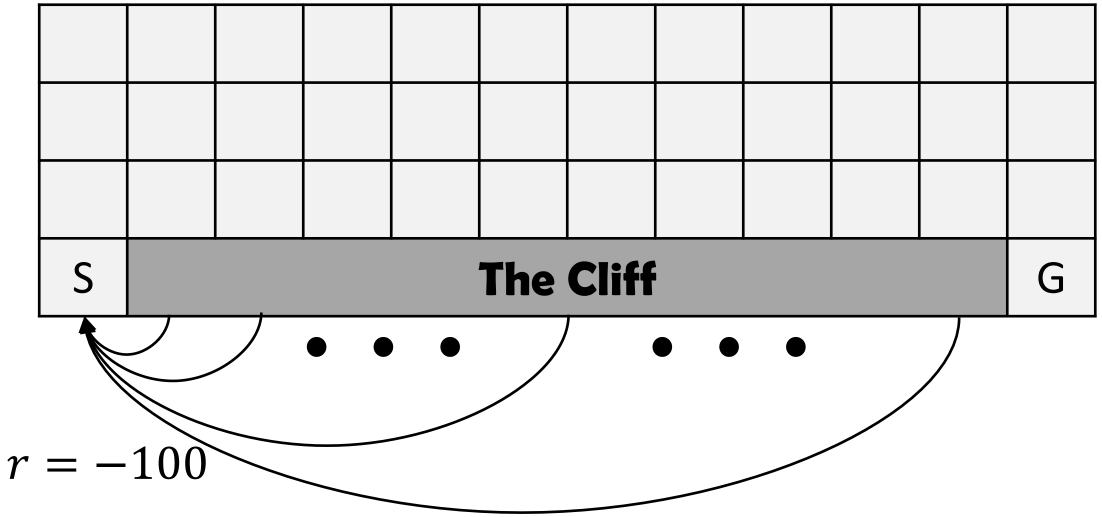
$$\pi(a|S_t) \leftarrow \begin{cases} 1 - \varepsilon + \varepsilon/|\mathcal{A}(S_t)| & \text{if } a = A^* \\ \varepsilon/|\mathcal{A}(S_t)| & \text{if } a \neq A^* \end{cases}$$

On-policy Vs. Off-policy

1. **On-policy methods,** are methods where you evaluate and improve the policy that is used to create the episodes.
    - To ensure all actions are selected we need to employ an ϵ-soft policy
2. Instead we can use **off-Policy methods.** In this case the policy we use to generate the episodes is different than the behavior we are trying to evaluate and improve.
    - Behavior policy $b(a|s)$ – The policy used to generate episodes
    - Estimation policy $\pi(a|s)$ – The policy being evaluated and improved.
3. This has some advantages:
    - Include On-policy as special case ($b(a|s) = \pi(a|s)$)
    - Estimation policy can be deterministic (e.g. greedy) and therefore optimal vs near optimal (ϵ−soft).
    - Can learn from observing the behavior of other non-learning agents (such as humans)
4. Disadvantages:
    - More complex and slower to converge.

Cliff walking

1. Need to go from S to G.
2. Actions are standard S,N,E,W
3. Reward is -1 for every move until goal is reached
4. In addition any move into "the cliff" results in a -100 reward and sends the agent back to the start.



Off Policy Prediction – Importance Sampling

1. In order to be able to estimate $\pi(a|s)$ from $\mathrm{b}(a|s)$ we require **coverage**. That is:
$$\pi(a|s) > 0 \quad \Rightarrow \quad b(a|s) > 0$$
   - Implies that $b(a|s)$ must be stochastic in actions not in $\pi(a|s)$
2. On the other hand $\pi(a|s)$ can be a deterministic greedy policy with respect to the current estimate of the action value function.
3. How to estimate action values for $\pi(a|s)$ from $\mathrm{b}(a|s)$?
   - **Importance sampling:** weigh returns according to the relative probability of a trajectory occurring under the target and behavior policies.
4. Probability of a trajectory:

$$P(A_t, S_{t+1}, A_{t+1}, \ldots, S_T \,|S_t, A_{t:T-1} \sim \pi)$$
$$= \pi(A_t|S_t)p(S_{t+1}|S_t, A_t) \times \pi(A_{t+1}|S_{t+1})p(S_{t+2}|S_{t+1}, A_{t+1}) \ldots p(S_T|S_{T-1}, A_{T-1})$$
$$= \prod_{k=t}^{T-1} \pi(A_k|S_k)p(S_{k+1}|S_k, A_k)$$

Off Policy Prediction – Importance Sampling

1. We then can define the relative probability of the trajectory under the target and behavior policies:

$$\rho_{t:T-1} = \frac{\prod_{k=t}^{T-1} \pi(A_k|S_k)p(S_{k+1}|S_k, A_k)}{\prod_{k=t}^{T-1} b(A_k|S_k)p(S_{k+1}|S_k, A_k)} = \frac{\prod_{k=t}^{T-1} \pi(A_k|S_k)}{\prod_{k=t}^{T-1} b(A_k|S_k)}$$

2. Using this we can calculate $V^\pi(s)$ given $V^b(s)$:

$$V^b(s) = \mathbb{E}_b(G_t|S_t = s)$$
$$V^\pi(s) = \mathbb{E}_b(\rho_{t:T-1} \times G_t|S_t = s)$$

3. Why does this work:

$$\mathbb{E}_b(\rho_{t:T-1} \times G_t|S_t = s) = \mathbb{E}_b\left(\frac{\prod_{k=t}^{T-1} \pi(A_k|S_k)}{\prod_{k=t}^{T-1} b(A_k|S_k)} \times G_t \middle| S_t = s\right) =$$

$$\sum_{\{trajectories\}} \frac{\prod_{k=t}^{T-1} \pi(A_k|S_k)}{\prod_{k=t}^{T-1} b(A_k|S_k)} \times G_t \times \prod_{k=t}^{T-1} b(A_k|S_k) = \sum \prod_{k=t}^{T-1} \pi(A_k|S_k) \times G_t = \mathbb{E}_\pi(G_t|S_t = s)$$

Off Policy Prediction – Importance Sampling

Ordinary importance sampling: $$V^\pi(s) = \frac{\sum_{t\in\mathcal{T}(s)} \rho_{t:T(t)-1} \times G_t}{|\mathcal{T}(s)|}$$

Weighted importance sampling: $$V^\pi(s) = \frac{\sum_{t\in\mathcal{T}(s)} \rho_{t:T(t)-1} \times G_t}{\sum_{t\in\mathcal{T}(s)} \rho_{t:T(t)-1}}$$

Where $T(t)$ is the time of the end of an episode which step s_t is in, $\mathcal{T}(s)$ is all the times we visit state s (or only first visits), and G_t are the returns from t to $T(t)$.

Use weighted importance sampling to ensure variance is finite.

- Ordinary importance sampling: unbiased, but variance is unbounded because ratios are unbounded
- Weighted importance sampling: biased, but bias converges to zero asymptotically and variance is bounded

Off Policy Prediction – Importance Sampling

1. Estimate the value of hitting for anything under 20, given a 50%-50% policy.
2. We do this just for one state: (13,2,1)
3. Use both ordinary and weighted importance sampling

$$\rho_{t:T-1} = \frac{\prod_{k=t}^{T-1} \pi(A_k|S_k)}{\prod_{k=t}^{T-1} b(A_k|S_k)}$$

$$V^\pi(s) = \frac{\sum_{t\in\mathcal{T}(s)} \rho_{t:T(t)-1} \times G_t}{|\mathcal{T}(s)|} \text{ or}$$

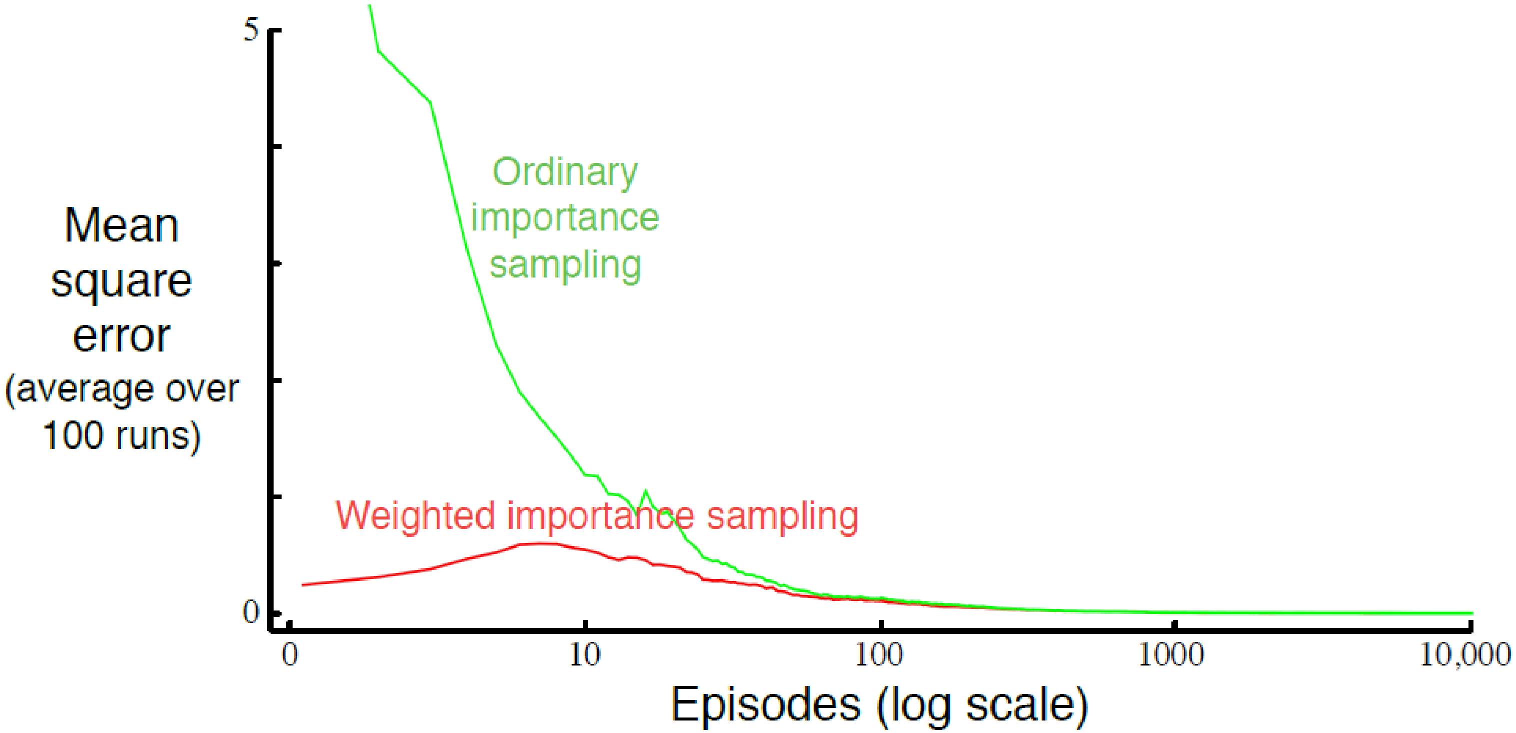
$$V^\pi(s) = \frac{\sum_{t\in\mathcal{T}(s)} \rho_{t:T(t)-1} \times G_t}{\sum_{t\in\mathcal{T}(s)} \rho_{t:T(t)-1}}$$

```
b(a | s) = [0.5 * H, 0.5 * S] for all S - hit or stand 50/50
Estimation policy pi(a | s) = hit under 20, stand otherwise
Run an episode to estimate: (13, 2, 1) -> let’s say we H -> 0 -> (18, 2, 1) -> H -> 0 -> (21, 2, 1) -> S -> 1 -> T
  returns[13, 2, 1] = [return * rho]
  Calculate rho = ( 1 * 1 * 1 ) / ( 0.5 * 0.5 * 0.5 ) = 8 therefore returns[13, 2, 1] = [ 1 * 8 ]
  Vpi(13, 2, 1) = avg(returns[13,2,1]) = 8
Another episode: (13, 2, 1) -> S -> -1 -> T
   rho = 0/0.5 = 0
   returns[13, 2, 1] = [ 8 * 1, 0 * 1 ]
   Vpi(13, 2, 1) = ( 8 + 0 ) / 2 = 4

For weighted importance sampling:
   need to store an array of Rhos, and we have Rhos[13,2,1] = [8, 0]
   V1(13, 2, 1) = 8 / 8 = 1
   V2(13, 2, 1) = (8 + 0)/(8 + 0)
```

Off Policy Prediction – Importance Sampling

1. Estimate the value of hitting for anything under 20, given a 50%-50% policy.
2. We do this just for one state: (13,2,1)
3. Use both ordinary and weighted importance sampling



$$\rho_{t:T-1} = \frac{\prod_{k=t}^{T-1} \pi(A_k|S_k)}{\prod_{k=t}^{T-1} b(A_k|S_k)}$$

$$V^{\pi}(s) = \frac{\sum_{t\in\mathcal{T}(s)} \rho_{t:T(t)-1} \times G_t}{|\mathcal{T}(s)|} \text{ or}$$

$$V^{\pi}(s) = \frac{\sum_{t\in\mathcal{T}(s)} \rho_{t:T(t)-1} \times G_t}{\sum_{t\in\mathcal{T}(s)} \rho_{t:T(t)-1}}$$

Importance Sampling – Incremental Implementation

1. Instead of storing all possible returns we can use similar methods to those described in Chapter 2.

$$NewEstimate \leftarrow OldEstimate + StepSize[NewData - OldEstimate]$$

2. Ordinary Sampling:
   - $V_{n+1} = V_n + \frac{1}{n}[\rho G_n - V_n]$

3. Weighted Sampling:
   - $V_n = \frac{\sum_{k=1}^{n-1} \rho_k G_k}{\sum_{k=1}^{n-1} \rho_k}, \quad n \geq 2$
   - Define: $C_{n+1} = C_n + \rho_{n+1}$
   - And therefore: $V_{n+1} = V_n + \frac{\rho_n}{C_n}[G_n - V_n]$

Off Policy MC Prediction

Off-policy MC prediction (policy evaluation) for estimating $Q \approx q_\pi$

Input: an arbitrary target policy π
Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
$\quad Q(s,a) \in \mathbb{R}$ (arbitrarily)
$\quad C(s,a) \leftarrow 0$

Loop forever (for each episode):
$\quad b \leftarrow$ any policy with coverage of π
$\quad$ Generate an episode following b: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
$\quad G \leftarrow 0$
$\quad W \leftarrow 1$
$\quad$ Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
$\quad\quad G \leftarrow \gamma G + R_{t+1}$
$\quad\quad C(S_t, A_t) \leftarrow C(S_t, A_t) + W$
$\quad\quad Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t,A_t)} [G - Q(S_t, A_t)]$
$\quad\quad W \leftarrow W \frac{\pi(A_t|S_t)}{b(A_t|S_t)}$
$\quad\quad$ If $W = 0$ then exit For loop

Off Policy MC Prediction

Off-policy MC control, for estimating $\pi \approx \pi_*$

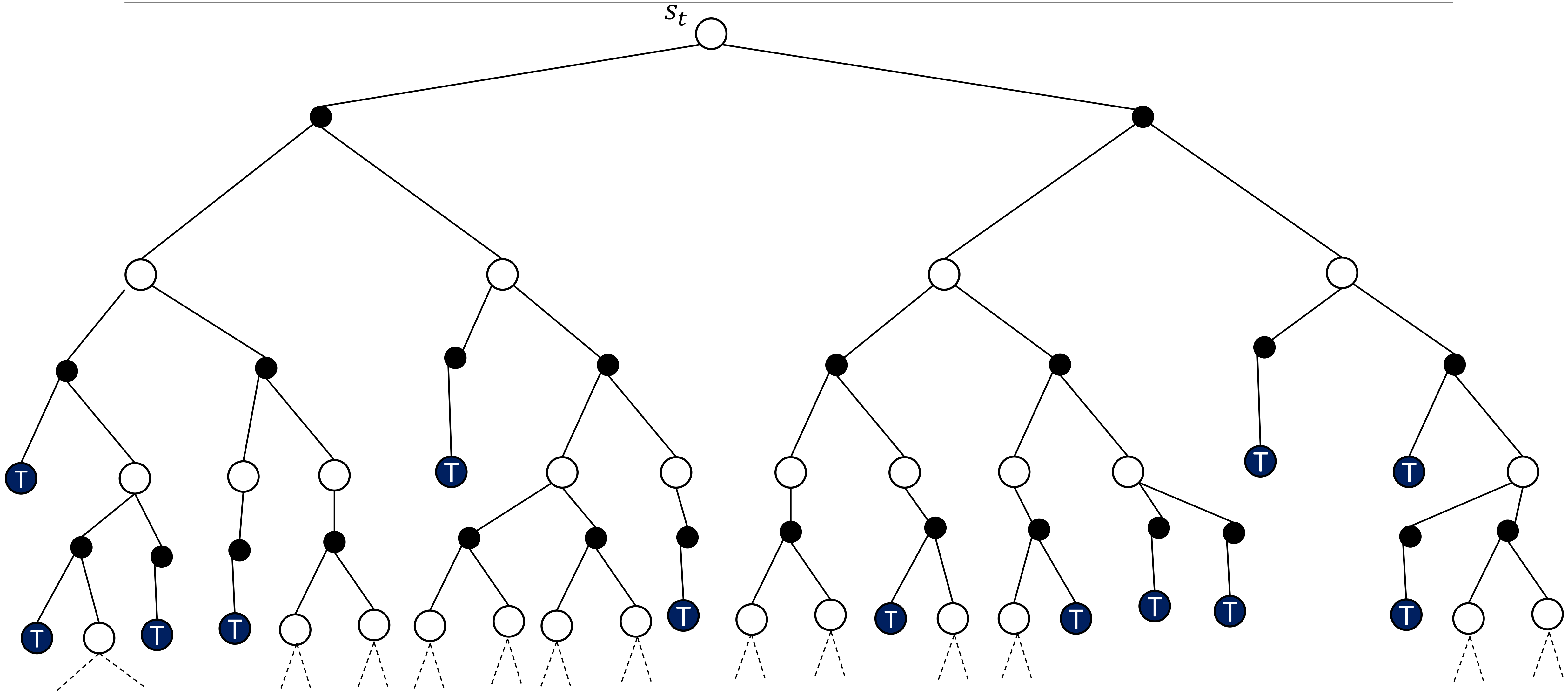
Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
- $Q(s,a) \in \mathbb{R}$ (arbitrarily)
- $C(s,a) \leftarrow 0$
- $\pi(s) \leftarrow \operatorname{argmax}_a Q(s,a)$ (with ties broken consistently)

Loop forever (for each episode):
- $b \leftarrow$ any soft policy
- Generate an episode using b: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
- $G \leftarrow 0$
- $W \leftarrow 1$
- Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
  - $G \leftarrow \gamma G + R_{t+1}$
  - $C(S_t, A_t) \leftarrow C(S_t, A_t) + W$
  - $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t,A_t)}[G - Q(S_t, A_t)]$
  - $\pi(S_t) \leftarrow \operatorname{argmax}_a Q(S_t, a)$ (with ties broken consistently)
  - If $A_t \neq \pi(S_t)$ then exit For loop
  - $W \leftarrow W \frac{1}{b(A_t|S_t)}$

Introduction to Temporal Difference Learning

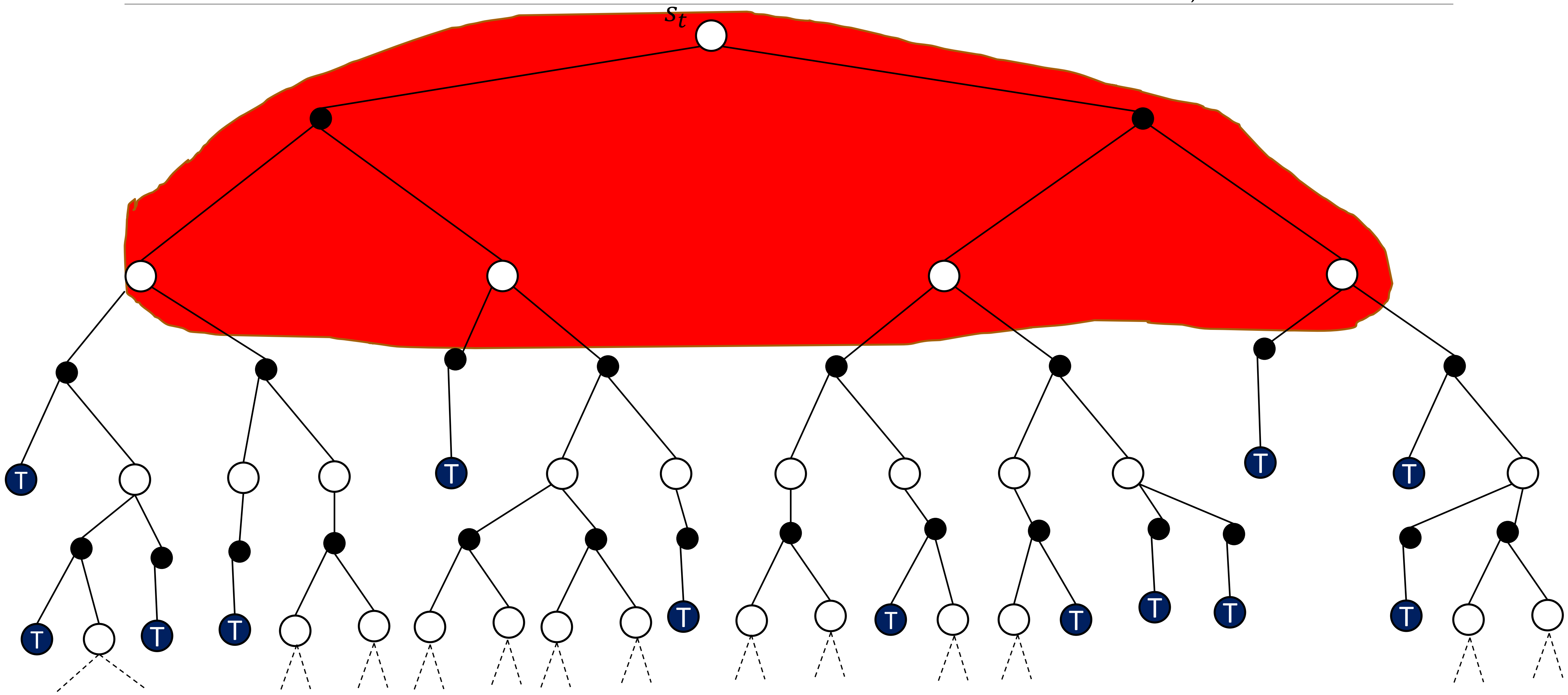
1. Perhaps the most central and novel idea to reinforcement learning.
   - Combination of ideas from DP and Monte Carlo
   - Learns without a model (like MC), bootstraps (like DP)
2. Both TD and Monte Carlo methods use experience to solve the prediction problem (policy evaluation).
3. A simple every-visit MC method may be expressed as:
   $$V^{\pi}(s_t) \leftarrow V^{\pi}(s_t) + \alpha\big(G_t - V^{\pi}(s_t)\big)$$
   - α can be $1/n$, or constant
4. Recall that in MC we need to wait until the end of the episode to update the value estimates.
5. The main idea in TD is to do an update every step.
6. Simplest TC method, TD(0):
   $$V^{\pi}(s) \leftarrow V^{\pi}(s) + \alpha\big(r_{t+1} + \gamma V^{\pi}(s_{t+1}) - V^{\pi}(s)\big)$$
7. Essentially, we are updating one guess based on another.

Backup Diagram



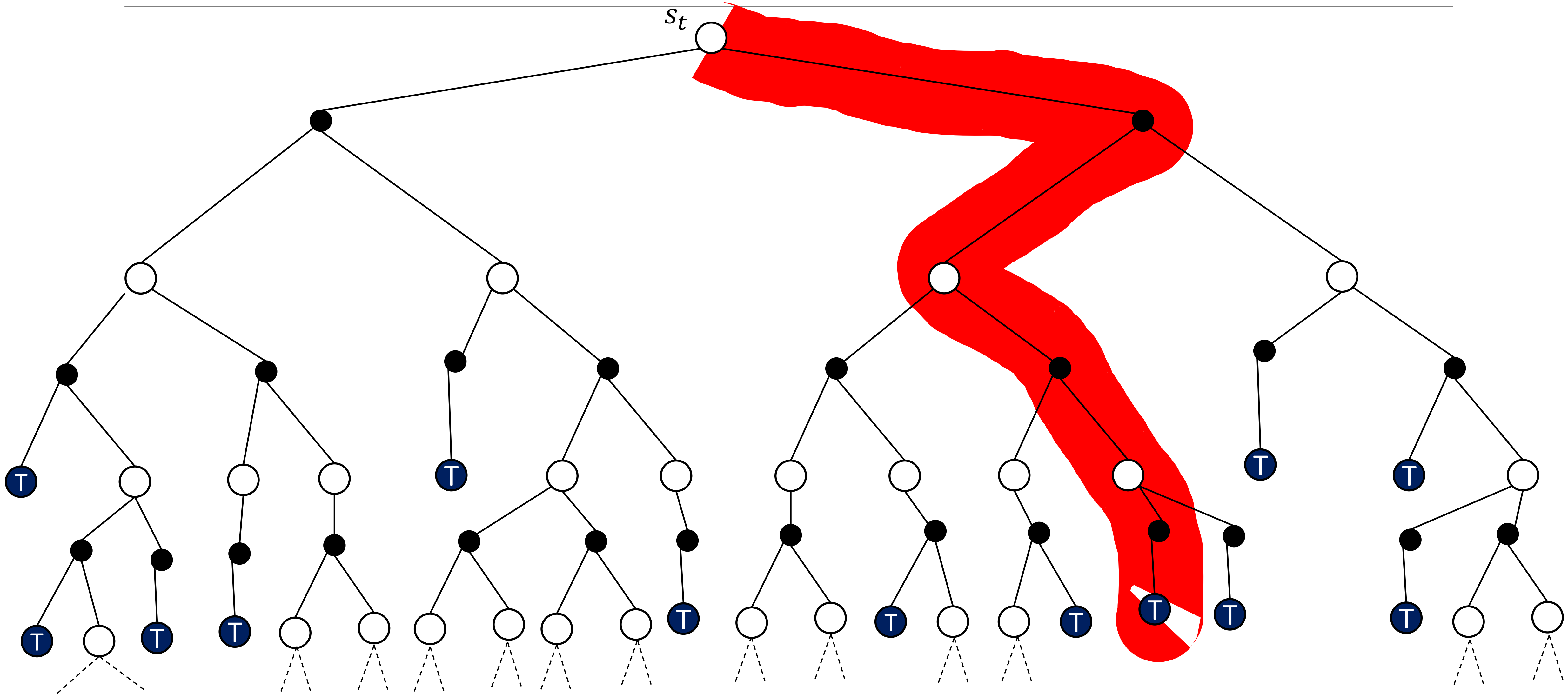
Backup Diagram – DP Update

$$V^{\pi}(s_t) = E_{\pi}[r_{t+1} + \gamma V^{\pi}(s_{t+1})] = \sum_{a} \pi(a|s_t) \sum_{s',r} p(s',r|s_t,a)(r + \gamma V^{\pi}(s'))$$



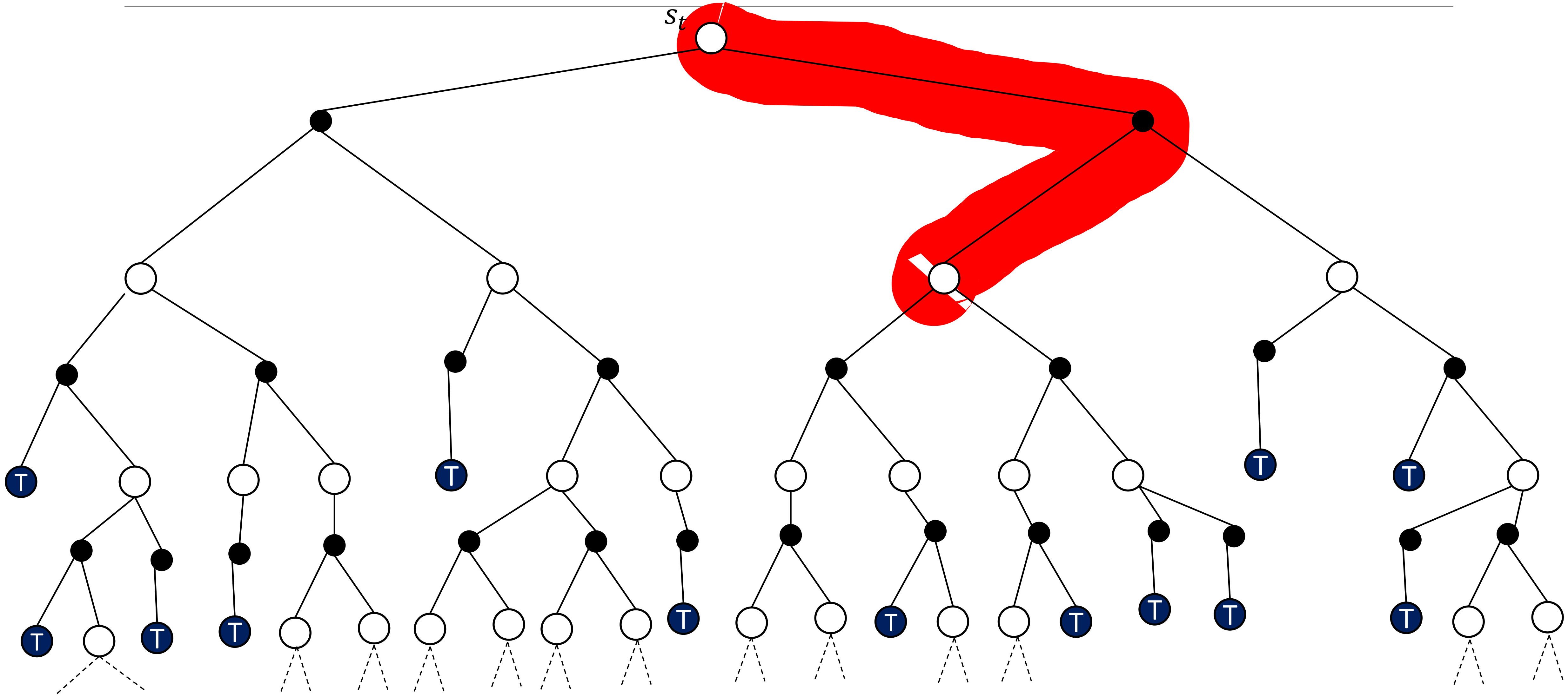
Backup Diagram – MC Update

$$V^{\pi}(s_t) \leftarrow V^{\pi}(s_t) + \alpha\big(G_t - V^{\pi}(s_t)\big)$$



$$V^{\pi}(s) \leftarrow V^{\pi}(s) + \alpha\big(r_{t+1} + \gamma V^{\pi}(s_{t+1}) - V^{\pi}(s)\big)$$

Backup Diagram – TD(0) Update



TD(0) for estimating V^{π}

Tabular TD(0) for estimating v_π

Input: the policy π to be evaluated
Algorithm parameter: step size $\alpha \in (0, 1]$
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(\mathit{terminal}) = 0$

Loop for each episode:
  Initialize S
  Loop for each step of episode:
    $A \leftarrow$ action given by π for S
    Take action A, observe R, S'
    $V(S) \leftarrow V(S) + \alpha\big[R + \gamma V(S') - V(S)\big]$
    $S \leftarrow S'$
  until S is terminal

TD methods bootstrap and sample

1. Bootstrapping: Update involves an estimate (a guess)
   - MC?
   - DP?
   - TD?
2. Sampling: update does not involve an expected value
   - MC?
   - DP?
   - TD?

DP

$$V^\pi(s_t) = E_\pi[r_{t+1} + \gamma V^\pi(s_{t+1})] = \sum_a \pi(a|s_t) \sum_{s',r} p(s', r|s_t, a)(r + \gamma V^\pi(s'))$$

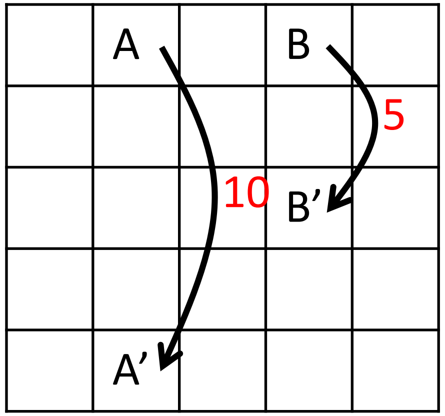
MC

$$V^\pi(s_t) \leftarrow V^\pi(s_t) + \alpha\big(G_t - V^\pi(s_t)\big)$$

TD

$$V^\pi(s) \leftarrow V^\pi(s) + \alpha\big(r_{t+1} + \gamma V^\pi(s_{t+1}) - V^\pi(s)\big)$$

Example: Grid World



$$\pi(a|s) = \{0.25,0.25,0.25,0.25\}\forall s$$

$$V_{k+1}^{\pi}(s) = \sum_{a} \pi(a|s) \sum_{s',r} p(s',r|s,a)[r + \gamma V_k^{\pi}(s')]$$

3.31	8.79	4.43	5.32	1.49
1.52	2.99	2.25	1.91	0.55
0.05	0.74	0.67	0.36	-0.40
-0.97	-0.44	-0.35	-0.59	-1.18
-1.86	-1.34	-1.23	-1.42	-1.97

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

V_0

-0.50	10.0	-0.25	5.00	-0.5
-0.25	0.00	0.00	0.00	-0.25
-0.25	0.00	0.00	0.00	-0.25
-0.25	0.00	0.00	0.00	-0.25
-0.50	-0.25	-0.25	-0.25	-0.5

V_1

1.47	9.78	3.07	5.00	0.34
-0.48	2.19	-0.06	1.07	-0.48
-0.42	-0.06	0.00	-0.06	-0.42
-0.48	-0.11	-0.06	-0.11	-0.48
-0.84	-0.48	-0.42	-0.48	-0.84

V_2

2.25	9.57	3.75	4.95	0.67
0.37	2.07	1.42	0.99	-0.13
-0.57	0.37	-0.05	0.12	-0.57
-0.66	-0.24	-0.14	-0.24	-0.66
-1.09	-0.66	-0.57	-0.66	-1.09

V_3

Example: Grid World

$\pi(a|s) = \{0.25, 0.25, 0.25, 0.25\} \forall s$

Why?

dynamic programming

3.31	8.79	4.43	5.32	1.49
1.52	2.99	2.25	1.91	0.55
0.05	0.74	0.67	0.36	-0.40
-0.97	-0.44	-0.35	-0.59	-1.18
-1.86	-1.34	-1.23	-1.42	-1.97

$$V^{\pi}(s) \leftarrow V^{\pi}(s) + \alpha\big(r_{t+1} + \gamma V^{\pi}(s_{t+1}) - V^{\pi}(s)\big)$$
$$\alpha = 0.1, \gamma = 0.9$$

3.05	9.00	5.65	5.59	1.77
2.20	3.25	3.12	2.94	0.65
0.99	1.31	1.15	0.61	-0.48
-0.76	-0.24	-0.13	-0.41	-0.89
-1.48	-1.13	-0.91	-1.29	-1.75

$(0,0)\ E$

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

V_0

$(0,1)\ N$

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

V_1

$(4,1)\ S$

0	1	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

V_2

$(4,1)\ N$

0	1	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	-0.1	0	0	0

V_3

$(3,1)\ S$

0	1	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	-0.09	0	0	0

V_4

0	1	0	0	0
0	0	0	0	0
0	0	0	0	0
0	-0.0081	0	0	0
0	-0.09	0	0	0

V_5