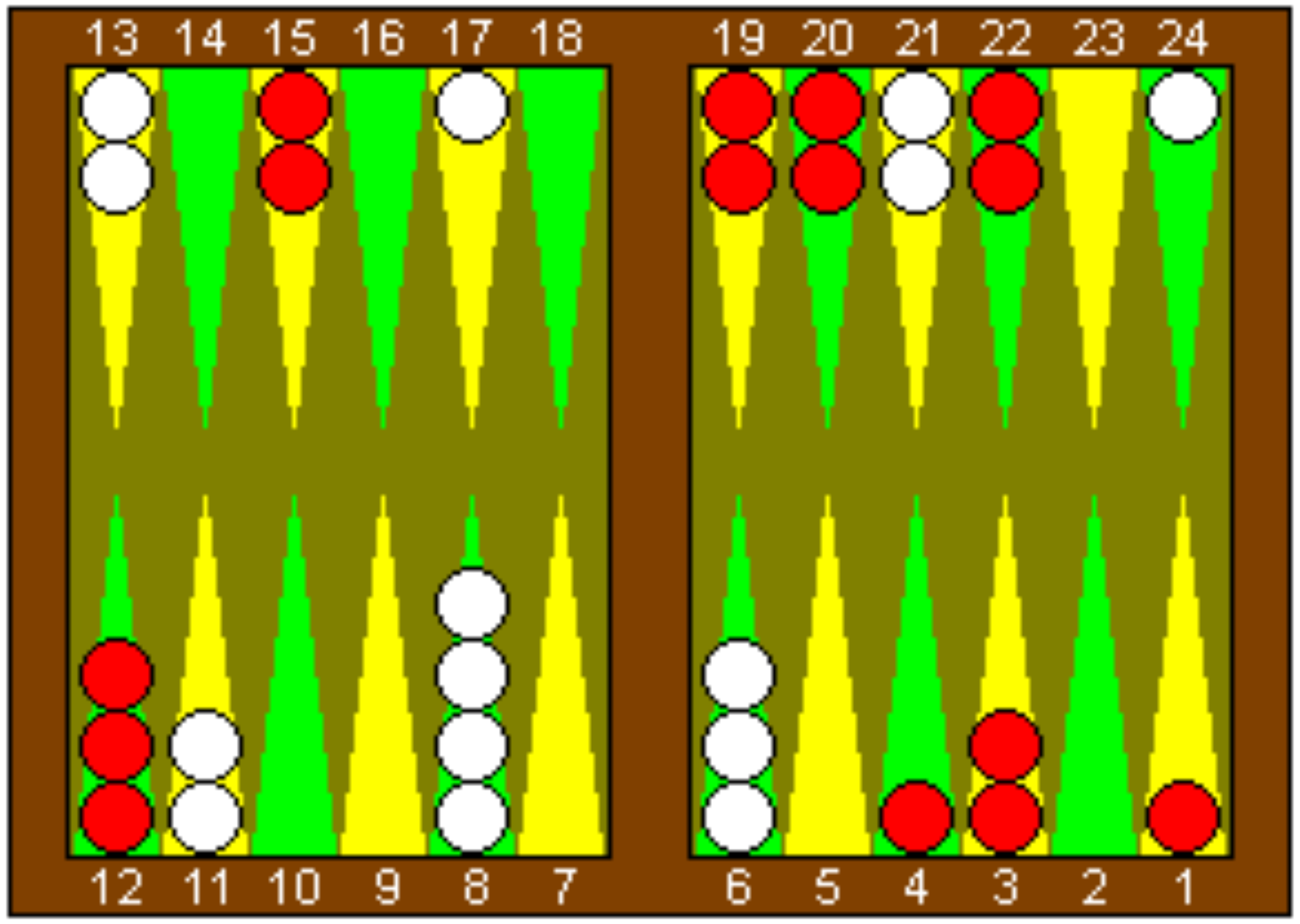
HSLU Lucerne University of Applied Sciences and Arts

Temporal Difference Learning: Summary



Information Technology
November 17, 2022

FH Zentralschweiz

TD Prediction

TD(0) methods update the state- or action-value function based on the estimates for the next state or state-action pair

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

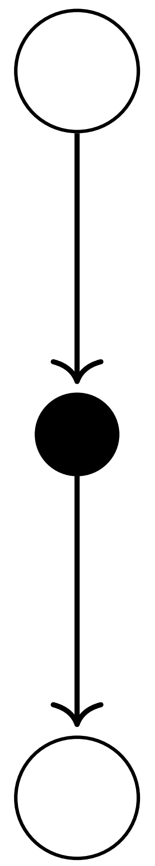
$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[\underbrace{R_{t+1} + \gamma Q(S_{t+1}, A_{t+1})}_{\text{Target}} - Q(S_t, A_t)]$$

TD(0) Prediction

TD(0) for estimating v_π

Input:
  the policy π to be evaluated
  step size $\alpha \in (0, 1]$
Initialize:
  $V(S)$ arbitrarily (except $V(\text{terminal} = 0)$

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



TD(0)

TD(0) Control

There are 3 TD(0) control methods which all use Generalized-Policy-Iteration (GPI)

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

Q-Learning

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

Expected Sarsa

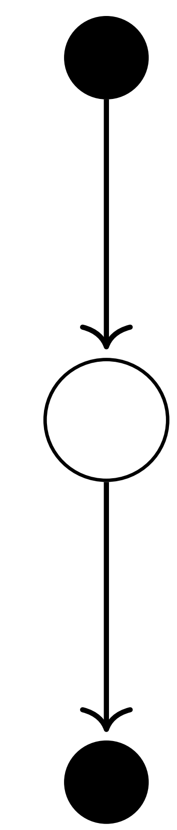
$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \sum_a \pi(a|S_{t+1}) Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)]$$

SARSA

Sarsa for estimating $Q \approx q_*$

Input:
  step size $\alpha \in (0, 1]$
  small $\epsilon > 0$
Initialize:
  $Q(s, a)$ for all $s \in \mathcal{S}^+, a \in \mathcal{A}$ arbitrarily (except $Q(\text{terminal}, \cdot) = 0$)

Loop for each episode:
  Initialize S
  Choose A from S using a policy derived from Q (e.g., ϵ-greedy)
  Loop for each step of the episode:
    Take action A, observe R, S'
    Choose A' from S' using a policy derived from Q (e.g., ϵ-greedy)
    $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma Q(S', A') - Q(S, A)]$
    $S \leftarrow S'$; $A \leftarrow$A'
  until S is terminal



Q-Learning

Q-learning for estimating $Q \approx q_*$

Input:
  step size $\alpha \in (0, 1]$
  small $\epsilon > 0$
Initialize:
  $Q(s, a)$ for all $s \in \mathcal{S}^+, a \in \mathcal{A}$ arbitrarily (except $Q(\text{terminal}, \cdot) = 0$)

Loop for each episode:
  Initialize S
  Loop for each step of the episode:
    Choose A from S using a 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

Targets of n-step returns

$$G_{t:t+1} \doteq R_{t+1} + \gamma V_t(S_{t+1})$$

$$G_{t:t+2} \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 V_{t+1}(S_{t+2})$$

$$G_{t:t+n} \doteq R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} + \gamma^n V_{t+n-1}(S_{t+n})$$

$$V_{t+n}(S_t) \doteq V_{t+n-1}(S_t) + \alpha[G_{t:t+n} - V_{t+n-1}(S_t)]$$