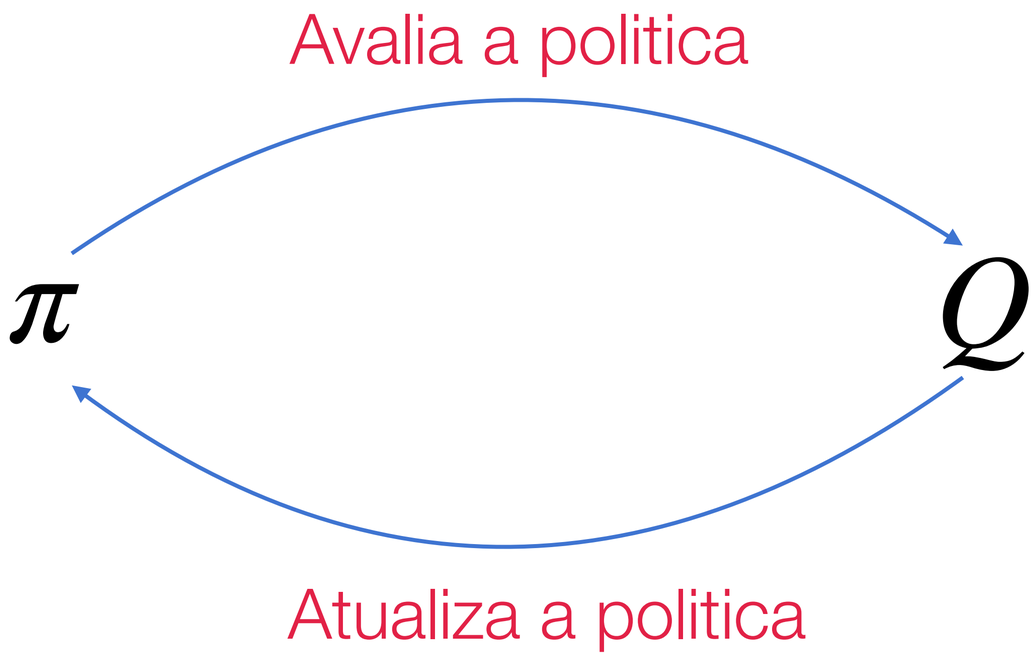
Métodos de Diferença temporal (TD)

O que são

- Base para a maioria dos métodos avançados de RL:
  - Tentativa e erro do MMC: aprende por experiência
  - Amostragem da PD: estimativas de valores $Q(s, a)$
  - Memória

O que são



O que são

Comparação

- Métodos de Monte-Carlo esperam ate o fim de um episódio para calcular G_t e atualizar $Q(s, a)$

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots + \gamma^{T-t-1} R_T$$

- Métodos de TD atualizam $Q(s, a)$ toda vez que um agente realiza uma ação

Alicerce de um Método de TD

- Como nos outros métodos, uma tabela de valores $q(s,a)$ é construída para cada par S_t, A_t

$$q_\pi(s,a) = \mathbb{E}[G_t \mid S_t = s, A_t = a] \qquad q_\pi(s,a) = \sum_{s',r} p(s', r \mid s, a)[r + \gamma v_\pi(s')]$$

$$q_\pi(s,a) = \sum_{s',r} p(s', r \mid s, a)[r + \gamma \sum_{a'} \pi(a' \mid s') q_\pi(s', a')]$$

Alicerce de um Método de TD

$$q_\pi(s,a) = \sum_{s',r} p(s',r \mid s,a)[r + \gamma \sum_{a'} \pi(a' \mid s') q_\pi(s',a')]$$

R_{t+1} S_{t+1} A_{t+1} Q

Alicerce de um Método de TD

- $Q_\pi(S_t, A_t)$ podem ser estimados por:

$$R_{t+1} + \gamma Q(S_{t+1}, A_{t+1})$$

$$R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)$$

Erro de Diferença temporal

Alicerce de um Método de TD

- Diferentes métodos possuem diferentes estimativas de $Q(S_t, A_t)$

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

Lembre-se do Método de Monte-Carlo com α constante

Alicerce de um Método de TD

- Reorganizando:

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

SARSA

Primeiro algoritmo de TD

- SARSA: $S_t, A_t, R_t, S_{t+1}, A_{t+1}$

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

- O SARSA é *on-policy*, só tem uma politica
- Como no MMC, usa a politica ϵ-gulosa para exploração

SARSA

Primeiro algoritmo de TD

Algorithm 1 SARSA

Input: α learning rate, ϵ random action probability, γ discount factor
$\pi \leftarrow \epsilon$-greedy policy w.r.t $Q(s, a)$
Initialize $Q(s, a)$ arbitrarily, with $Q(terminal, \cdot) = 0$
for episode $\in 1..N$ **do**
    Reset the environment and observe S_0
    $A_0 \sim \pi(S_0)$
    for $t \in 0..T-1$ **do**
        Execute A_t in the environment and observe S_{t+1}, R_{t+1}
        $A_{t+1} \sim \pi(S_{t+1})$
        $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)\right]$
    end for
end for
Output: Near optimal policy π and action values $Q(s, a)$

SARSA

Primeiro algoritmo de TD

Abra o Notebook SARSA.ipynb

Q-Learning

Segundo Algoritmo de TD

- O Q-Learning é *off-policy*, possui uma politica exploratória π_b
- Funciona exatamente como o SARSA, mas usa π_b para gerar uma trajetória

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

$$A_{t+1} = \pi(S_{t+1}) = \text{argmax}_a Q(S_{t+1}, a)$$

Q-Learning

Segundo Algoritmo de TD

Algorithm 2 Q-Learning

Input: α learning rate, γ discount factor
$\pi \leftarrow$ greedy policy w.r.t $Q(s, a)$
$b \leftarrow$ exploratory policy with coverage of π
Initialize $Q(s, a)$ arbitrarily, with $Q(terminal, \cdot) = 0$
for episode $\in 1..N$ **do**
    Reset the environment and observe S_0
    for $t \in 0..T-1$ **do**
        $A_t \sim b(S_t)$
        Execute A_t in the environment and observe S_{t+1}, R_{t+1}
        $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma\pi(S_{t+1}) - Q(S_t, A_t)\right]$
    end for
end for
Output: Approximately optimal policy π and action values $Q(s, a)$

Q-Learning

Segundo algoritmo de TD

Abra o Notebook QLearning.ipynb