RL aula 3

Previsão e Controle sem Modelo

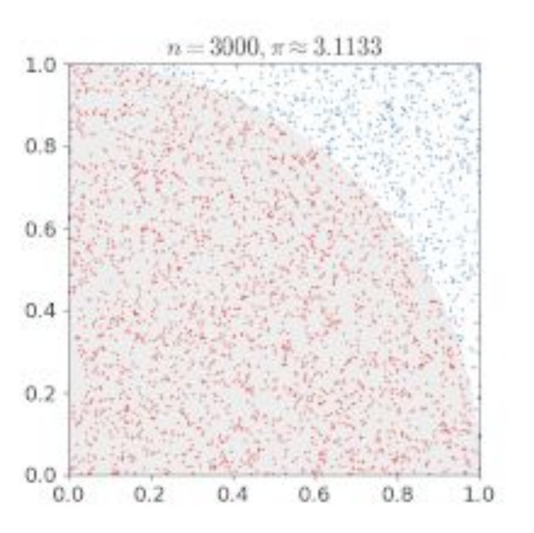
Plano de hoje

1. Model-free Prediction
    a. Monte Carlo
    b. Temporal Difference
2. Model-free Control
    a. On-policy
    b. Off-policy

Métodos de Monte Carlo

- Visão Geral
- Aplicado a MDPs

Visão Geral



- Usar aleatoriedade para resolver problemas
  - Até os que são a priori determinísticos
- Baseado na Lei dos Números Grandes
- Combinamos informação de amostras para aproximar soluções de problemas
  - Aproximar π
  - Aproximar integrais

Monte Carlo para MDPs

- Podemos usar para Prediction e para Control
- Aprende diretamente a partir de episódios de experiências
- Aprende a partir de episódios completos
  - Precisa que o MDP seja episódico
- É model-free

Monte-Carlo Policy Evaluation

- Queremos encontrar v_π a partir dos episódios
- Mas lembre que v é o valor esperado do retorno
- E que podemos calcular o retorno:

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

- Então podemos usar a média amostral para aproximar v

Versões

- Todas começam com a geração de um episódio $S_1, A_1, R_2, \ldots, S_k \sim \pi$
- Every Visit
  - Toda vez que o estado aparece no episódio atualizamos:

$$\text{Returns}(S_t) \leftarrow \text{Returns}(S_t) + G_t$$
$$N(S_t) \leftarrow N(S_t) + 1$$

- First Visit
  - Fazemos a atualização apenas na primeira vez
- No fim $V(s) \leftarrow \frac{\text{Returns}(s)}{N(s)}$ for all $s \in \mathcal{S}$
- Incremental Average

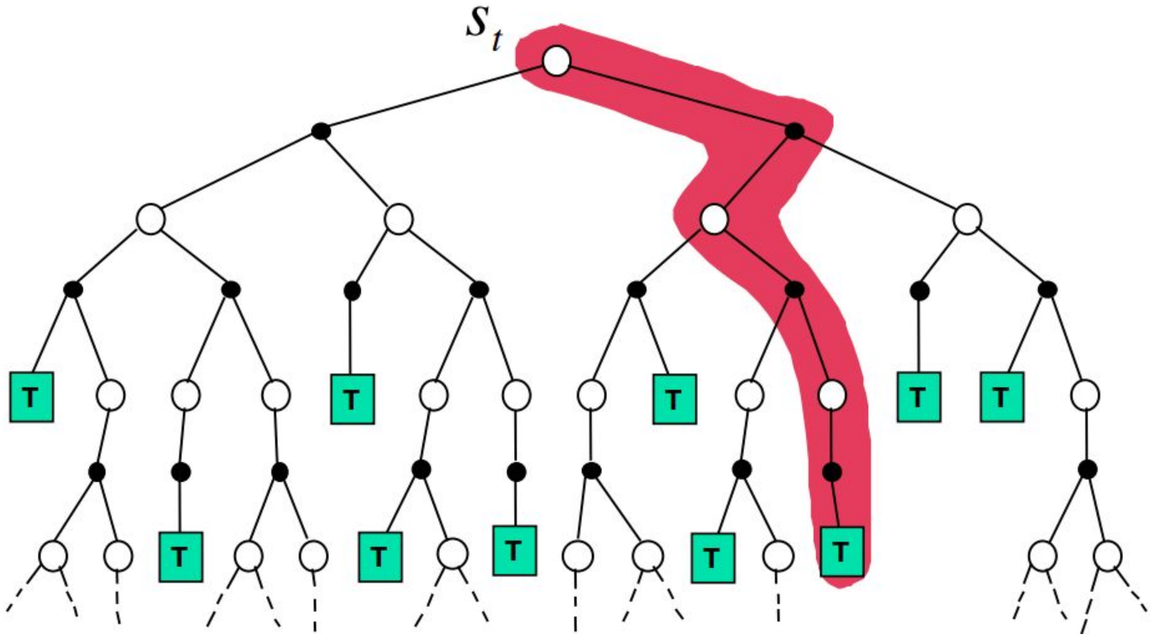
$$V(S_t) \leftarrow V(S_t) + \frac{1}{N(S_t)}(G_t - V(S_t))$$

  - Nos permite modificar o MC para problemas não estacionários

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

Diagrama de Backup do MC

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



Temporal-Difference Learning

- Model-free
- Aprende online através das experiências
- Usa bootstraping: Aprende a partir de episódios incompletos
- Aproxima a solução de DP usando amostragem

$$v_\pi(s) = \mathbb{E}_\pi \left[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s \right]$$

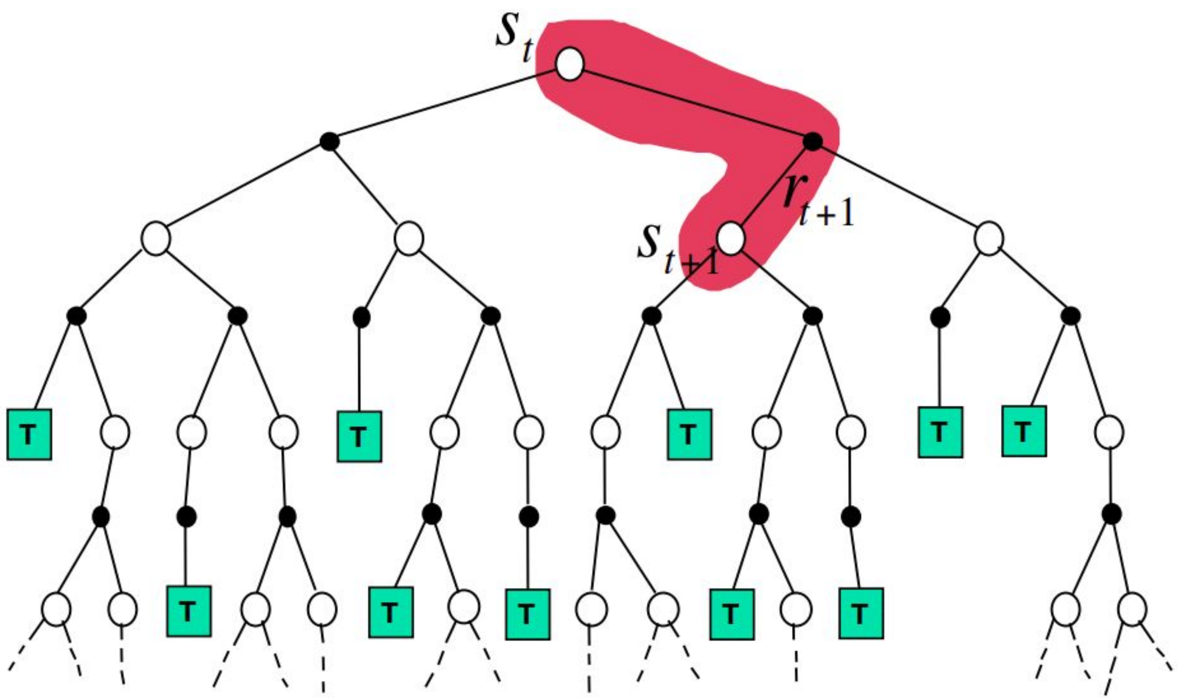
- "Updates a guess towards a guess"

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

- Disso temos:
    - TD Target: $R_{t+1} + \gamma V(S_{t+1})$
    - TD Error: Target - $V(S_t) = \delta_t$

Diagrama de Backup do TD(0)

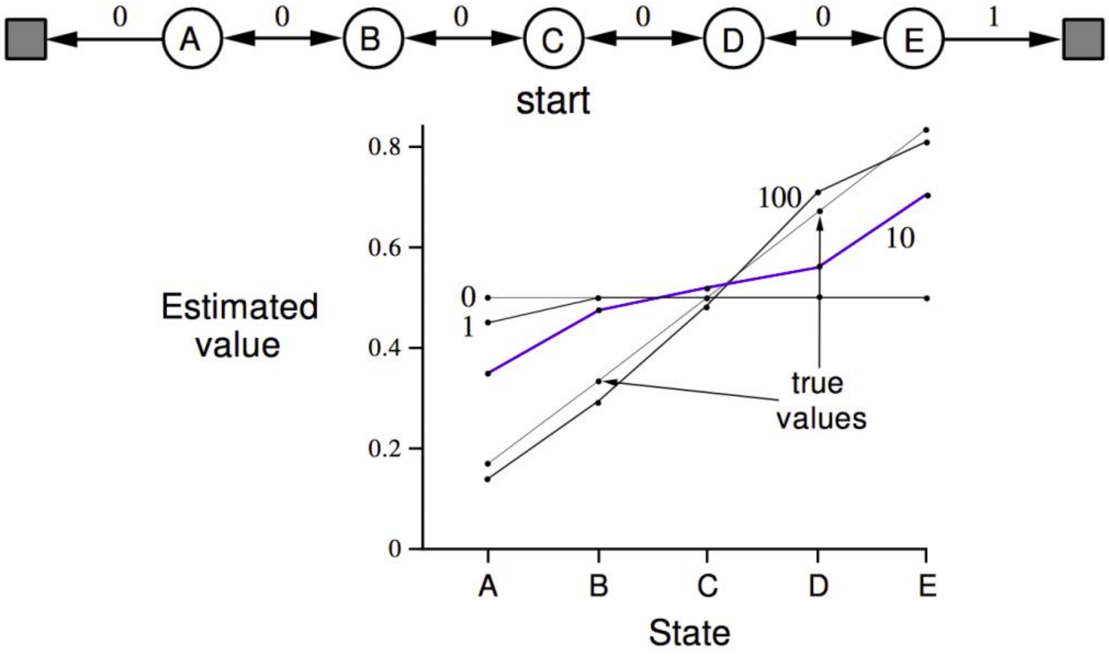
$$V(S_t) \leftarrow V(S_t) + \alpha \left(R_{t+1} + \gamma V(S_{t+1}) - V(S_t) \right)$$



Comparação

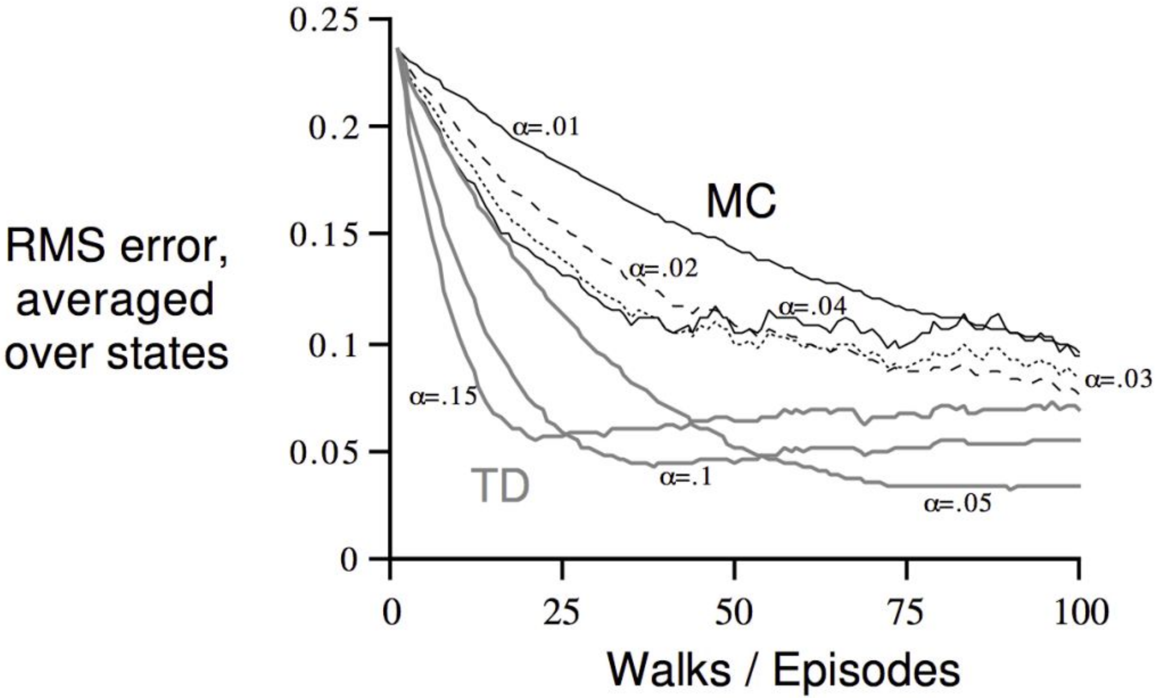
- TD aprende antes de saber o resultado final
- TD aprende mesmo sem o resultado final
  - Funciona em non-terminating MDPs
  - Pode aprender a partir de sequências incompletas
- Monte Carlo tem alta variância e 0 vício
  - G é um unbiased estimator de V
  - Boa convergência
  - Não é muito sensível ao valor inicial
- TD tem baixa variância e um pouco de viés
  - G depende de vários passos com eventos aleatório, TD-target depende de um passo apenas
  - Em geral mais eficiente que MC
  - Mais sensível a valores iniciais

Exemplo da Random Walk



Fonte: David Silver, 4th lecture

Exemplo da Random Walk: Comparação



Fonte: David Silver, 4th lecture

Batch MC e TD

- Ambos os métodos convergem para v com experiência infinita, mas o que rola se tivermos um batch finito?
- Vamos ter um conjunto de episódios, pegar uma amostra deles e aplicar TD(0) ou MC

Exemplo A, B

2 Estados, gamma = 0, 8 eps de exp

(A, 0) -> (B, 0)

(B, 1)

(B, 1)

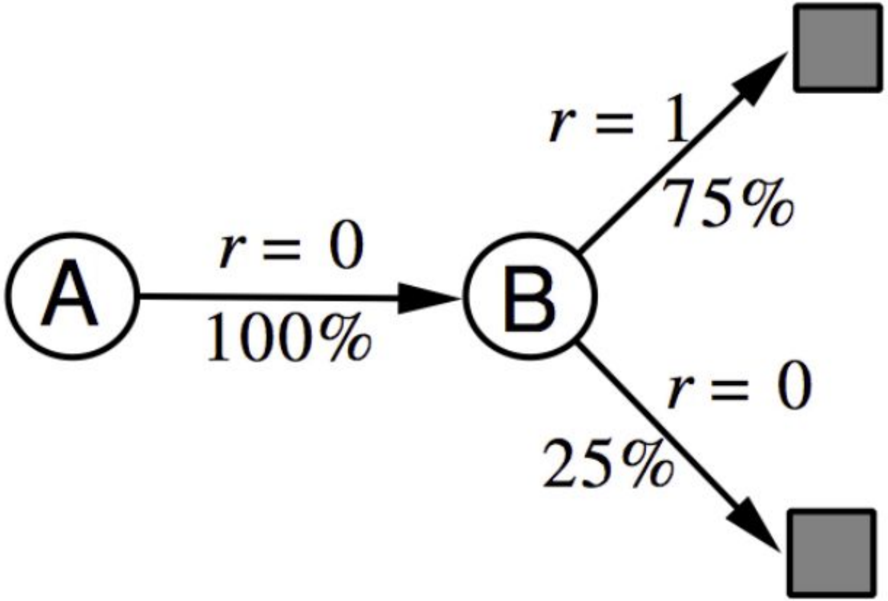
(B, 1)

(B, 1)

(B, 1)

(B, 1)

(B, 0)



Batch MC e TD

- Ambos os métodos convergem para v com experiência infinita, mas o que rola se tivermos um batch finito?
- Vamos ter um conjunto de episódios, pegar uma amostra deles e aplicar TD(0) ou MC
- Monte Carlo converge para a solução de menor MSE

$$\sum_{k=1}^{K} \sum_{t=1}^{T_k} \left(G_t^k - V(s_t^k) \right)^2$$

- TD(0) converge para a solução do MDP de maior verossimilhança

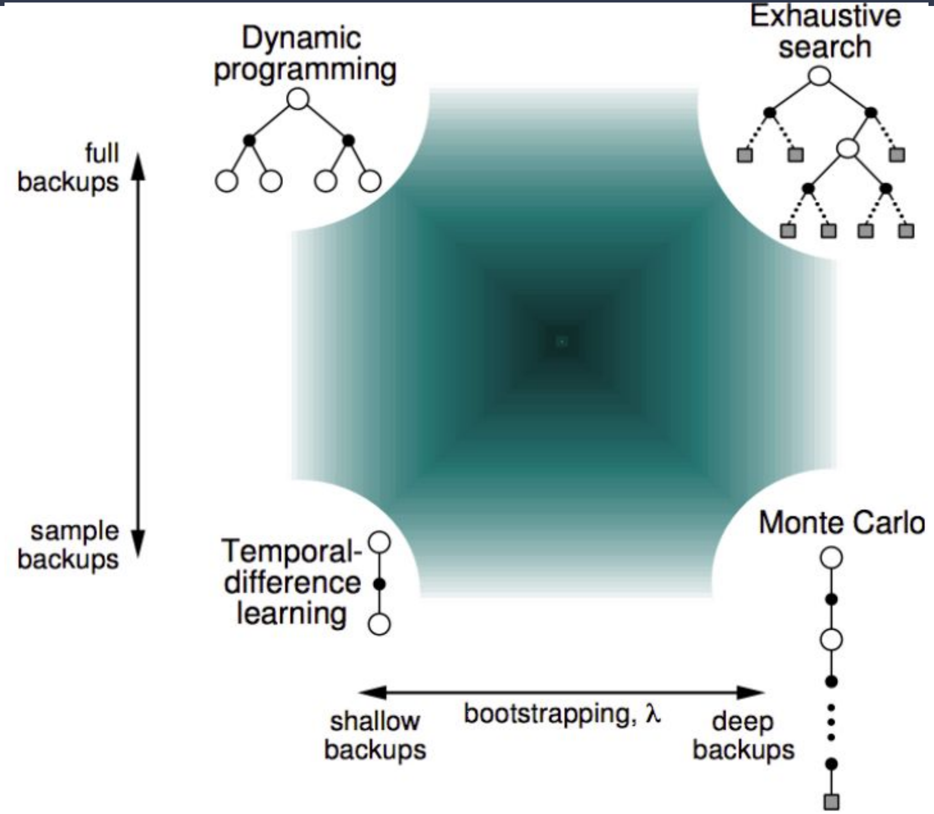
$$\hat{\mathcal{P}}_{s,s'}^{a} = \frac{1}{N(s,a)} \sum_{k=1}^{K} \sum_{t=1}^{T_k} \mathbf{1}(s_t^k, a_t^k, s_{t+1}^k = s, a, s')$$

$$\hat{\mathcal{R}}_{s}^{a} = \frac{1}{N(s,a)} \sum_{k=1}^{K} \sum_{t=1}^{T_k} \mathbf{1}(s_t^k, a_t^k = s, a) r_t^k$$

Comparação 2

- TD se aproveita da propriedade de Markov
  - Mais eficiente em MDPs
- MC não usa a propriedade de Markov
  - Melhor quando o problema não é um MDP
- Bootstrapping: Usa um estimativa no Update
  - DP
  - TD(0)
- Sampling: Usa uma amostra no update
  - TD(0)
  - MC

Visão Unificada de Reinforcement Learning



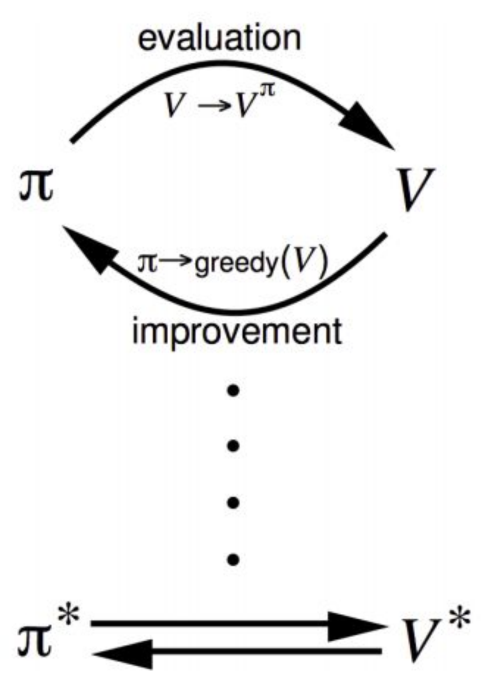
Model-Free Control

- On Policy vs Off Policy
- Monte Carlo Policy Iteration
- Exploration Strategies
- SARSA
- Off policy

On-Policy vs Off-Policy

- On-policy learning
  - Aprender sobre π usando experiências seguindo π
- Off-policy
  - Aprender sobre π usando experiências de π'

Generalised Policy Iteration



- Policy Evaluation: Estimar v
- Policy Improvement: Gerar uma policy melhor que a anterior

Monte Carlo Policy Iteration

- Policy Evaluation: Estimar v usando Monte Carlo
- Policy Improvement: Agir gulosamente em relação a v

- Greedy Policy Improvement com o v requer um modelo do MDP

$$\pi'(s) = \underset{a \in \mathcal{A}}{\operatorname{argmax}} \, \mathcal{R}_s^a + \mathcal{P}_{ss'}^a V(s')$$

- Podemos usar o q:

$$\pi'(s) = \underset{a \in \mathcal{A}}{\operatorname{argmax}} \, Q(s, a)$$

Estratégias de Exploração

- e-greedy
- GLIE

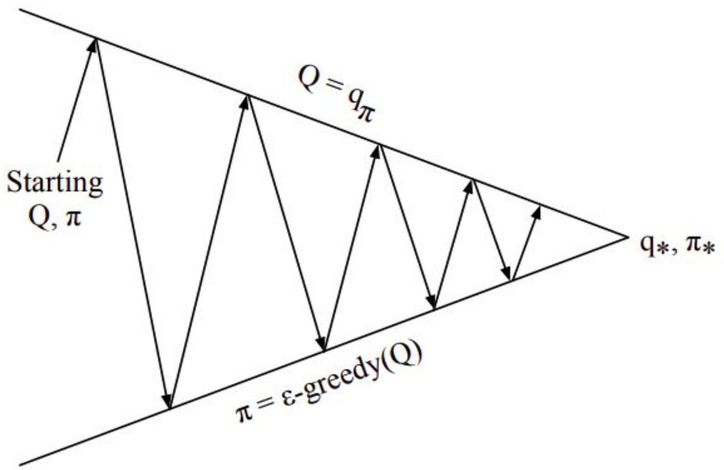
ε-greedy

- Ideia mais simples para garantir exploração
- Com probabilidade 1 - ε escolha gulosa
- Com probabilidade ε escolher uma ação aleatória

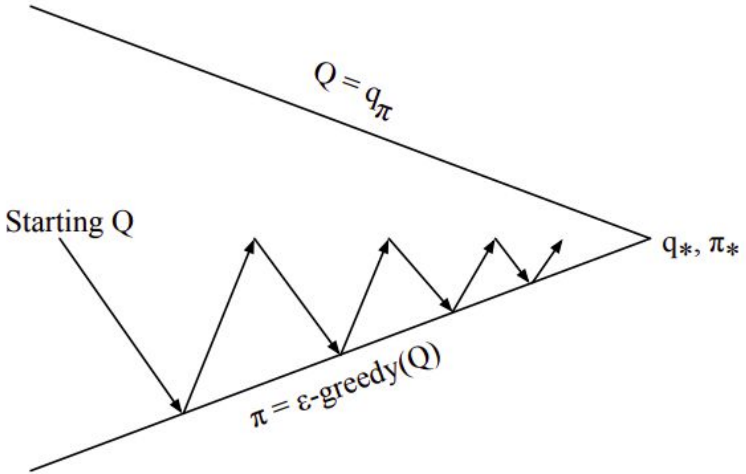
$$\pi(a|s) = \begin{cases} \epsilon/m + 1 - \epsilon & \text{if } a^* = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, Q(s,a) \\ \epsilon/m & \text{otherwise} \end{cases}$$

MC Policy Iteration and Control

Policy Iteration



Control

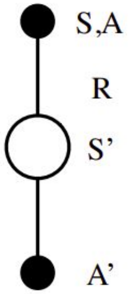


Greedy in the limit with infinite exploration

- Também testa todas as ações infinitamente
- A policy converge para uma policy gulosa
- Efetivamente ε-greedy com decaimento do ε

E TD?

- Relembrando as vantagens de TD:
  - Menor variância
  - Online
  - Lida com sequências incompletas
- Usar TD em vez de MC (Sarsa)



$$Q(S,A) \leftarrow Q(S,A) + \alpha \left(R + \gamma Q(S',A') - Q(S,A)\right)$$

SARSA

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

- Convergência garantida para alphas que são uma sequência de Robbins-Monro
  - Soma diverge
  - Soma dos quadrados converge

Off-Policy Learning

- Policy evaluation em π' enquanto segue π
- π' é target policy, π é behavior policy
- Qual a relevância?
  - Aprender observando humanos ou outros agentes
  - Reutilizar experiência de policies antigas
  - Aprender sobre a policy ótima enquanto usa uma policy exploradora
  - Aprender sobre várias policies enquanto segue apenas uma

Importance Sampling

- Estimar a esperança usando outra distribuição

$$\mathbb{E}_{X\sim P}[f(X)] = \sum P(X)f(X)$$
$$= \sum Q(X)\frac{P(X)}{Q(X)}f(X)$$
$$= \mathbb{E}_{X\sim Q}\left[\frac{P(X)}{Q(X)}f(X)\right]$$

Importance Sampling

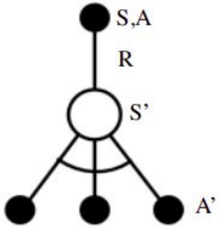
- Monte Carlo
- Pode aumentar muito a variância

$$G_t^{\pi/\mu} = \frac{\pi(A_t|S_t)}{\mu(A_t|S_t)} \frac{\pi(A_{t+1}|S_{t+1})}{\mu(A_{t+1}|S_{t+1})} \cdots \frac{\pi(A_T|S_T)}{\mu(A_T|S_T)} G_t$$

- TD
- Variância bem menor que MC

$$V(S_t) \leftarrow V(S_t) + \alpha \left(\frac{\pi(A_t|S_t)}{\mu(A_t|S_t)} (R_{t+1} + \gamma V(S_{t+1})) - V(S_t) \right)$$

Q-learning



$$Q(S, A) \leftarrow Q(S, A) + \alpha \left(R + \gamma \max_{a'} Q(S', a') - Q(S, A) \right)$$

- Aprender Q off-policy e sem importance sampling
- Escolhemos a próxima ação A_{t+1} usando μ
- Mas consideramos uma A' alternativa usando π
- E fazemos o update de Q usando A'

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left(R_{t+1} + \gamma Q(S_{t+1}, A') - Q(S_t, A_t) \right)$$

- Tomar target gulosa
- Mas behavior exploratória (GLIE ou epsilon greedy)
- "SARSAMax"

Recursos úteis

- Dissecting RL Monte Carlo
- Dissecting RL TD
- Lilian Weng's A (Long) Peek into Reinforcement Learning
- DeepMind RL course
- Stack Exchange: Evey-Visit vs First-Visit