Aprendizado por Reforço

AULA - 2

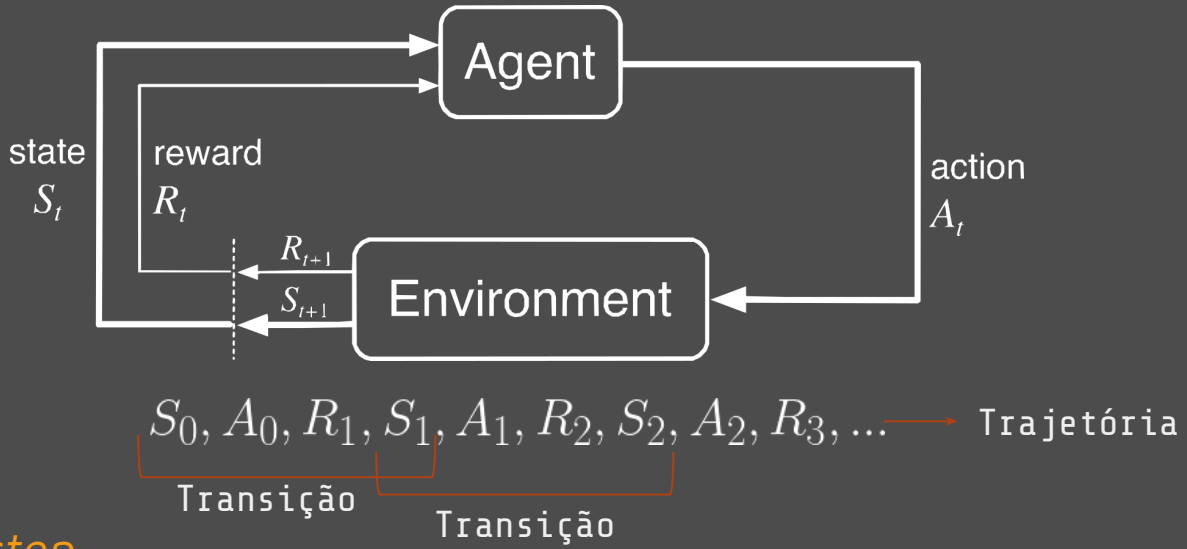
Processos de Decisão de Markov (MDP)

Retrospectiva do último episódio

Retrospectiva do último episódio

- Agente
  - Processa observações, escolhe ações, e aprende
- Ambiente
  - Elemento com o qual o agente interage (tudo que não é o agente)
- Observação
  - Descrição do estado do ambiente (pode ser parcial)
- Ação
  - Forma do agente interagir com o ambiente
- Recompensa
  - Diz o quão bom é o estado do ambiente (guia o aprendizado)

Relação Agente e Ambiente



- Passo/*timestep*
- Transições
- Episódios (tamanho T)

Notações Importantes

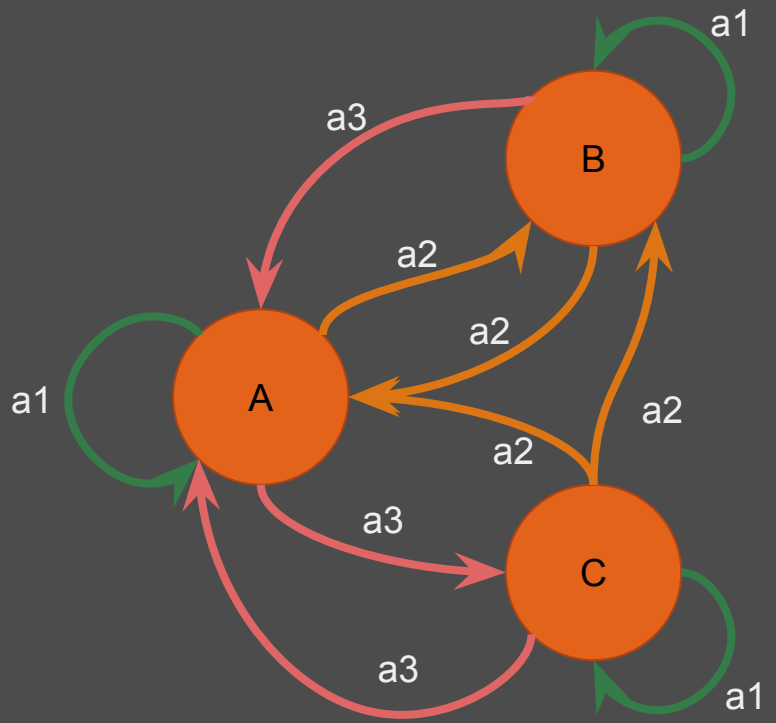
- Ação $a \in \mathcal{A}$
  - ação no timestep t: a_t
  - Espaço de Ações: $\mathcal{A}$ ex: [cima, baixo, esq., dir.]
- Estado $s \in \mathcal{S}$
  - estado no timestep t: s_t
  - Espaço de Estados: $\mathcal{S}$ ex: (leituras de um array de sensores)
- Recompensa $r \in \mathcal{R}$
  - recompensa no timestep t: r_t
  - Domínio da função de Recompensa: $\mathcal{R}$

Definindo Matematicamente o Ambiente

- Probabilidade de transição
- Probabilidade do próximo estado ser s' e a recompensa r dado que o estado anterior foi s e a ação a

$$p(s', r|s, a)$$

- Dinâmicas do ambiente
- O próximo estado depende APENAS do estado anterior e da ação tomada



	a1	a2	a3
A	P(A) = 1.0 P(B) = 0.0 P(C) = 0.0	P(A) = 0.0 P(B) = 1.0 P(C) = 0.0	P(A) = 0.0 P(B) = 0.0 P(C) = 1.0
B	P(A) = 0.0 P(B) = 1.0 P(C) = 0.0	P(A) = 1.0 P(B) = 0.0 P(C) = 0.0	P(A) = 1.0 P(B) = 0.0 P(C) = 0.0
C	P(A) = 0.0 P(B) = 0.0 P(C) = 1.0	P(A) = 0.5 P(B) = 0.5 P(C) = 0.0	P(A) = 1.0 P(B) = 0.0 P(C) = 0.0

Propriedade Markoviana

O passado é irrelevante dado o presente

A informação necessária para tomar uma decisão está completamente presente na representação do estado

Definindo Matematicamente o Agente

- Política de decisão estocástica
- Distribuição de probabilidade de ações a dado o estado s

$$\pi(a|s)$$

- Função de valor seguindo a política pi

$$V_\pi(s) \quad Q_\pi(s,a)$$

Qual o objetivo do Aprendizado por Reforço?

Maximizar o sinal de recompensa ao longo do tempo

O que é Recompensa Atrasada?

Quando há relação de causalidade entre uma ação em um passo t e uma recompensa em um passo t+n

O que é Recompensa ao longo do tempo?

- Retorno

$$R_t = r_{t+1} + r_{t+2} + \cdots + r_T$$

$$R_t = r_{t+1} + R_{t+1}$$

- T é o passo final de um MDP finito

E se o MDP for infinito?

- Fator de Desconto

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

para: $0 \leq \gamma < 1$

$$\sum_{k=0}^{\infty} \gamma^k = \frac{1}{1-\gamma}$$

Usando o Fator de Desconto... Sempre

- Noção temporal na recompensa
- “Punir” ao demorar para adquirir uma recompensa boa

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

$$R_t = r_{t+1} + \gamma R_{t+1}$$

Função de Valor

- Função de valor estima a expectativa do Retorno

$$V_\pi(s) = \mathbb{E}_\pi[R_t|s_t = s]$$

$$Q_\pi(s, a) = \mathbb{E}_\pi[R_t|s_t = s, a_t = a]$$

- Estando subordinada à política, ela pode ser uma forma de avaliar a política

Equação de Bellman

$$V_\pi(s) = \mathbb{E}_\pi[R_t | s_t = s]$$

$$V_\pi(s) = \mathbb{E}_\pi[r_{t+1} + \gamma R_{t+1} | s_t = s]$$

$$= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r|s, a)[r_{t+1} + \gamma \mathbb{E}_\pi[R_{t+1}|s_t = s]]$$

$$= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r|s, a)[r_{t+1} + \gamma V_\pi(s')]$$

Função de Valor Ótima

- A melhor função de valor estima o retorno dos estados perfeitamente, inclusive com as dinâmicas do ambiente consideradas.

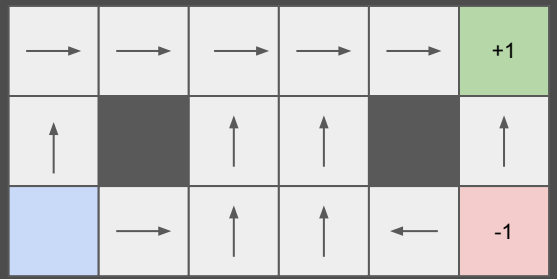
- *State value function* $V_*(s)$

- *State-action value function* $Q_*(s, a)$

Política Ótima

- A melhor política sempre escolhe os estados com maior retorno.

$$\pi_*$$

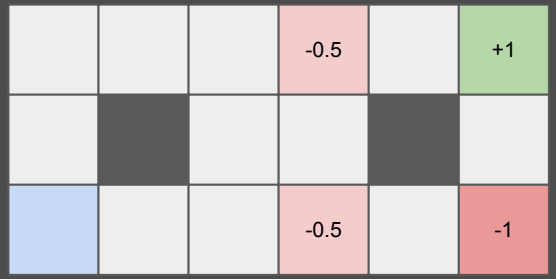


Premissa dos Algoritmos *Value-Based*

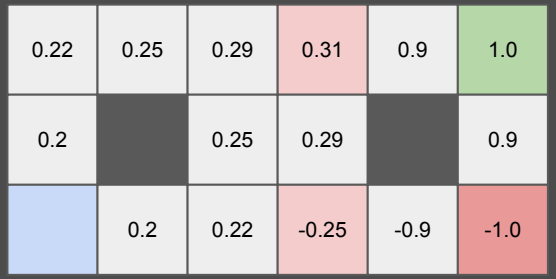
- A melhor política segue e resulta na melhor função de valor.
- Tendo a melhor função de valor, tem-se a melhor política.

$$V_*(s) = \max_{a \in \mathcal{A}} Q_{\pi_*}(s, a)$$

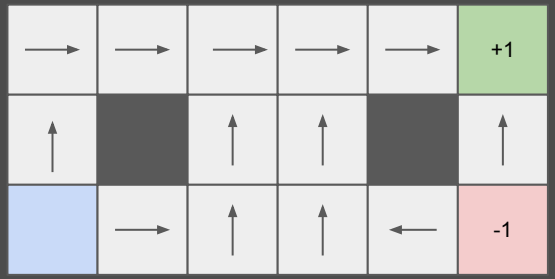
A melhor política segue a melhor função de valor



Recompensas



Valores



Política

Equação de Bellman para V ótimo

$$
\begin{aligned}
V_*(s) &= \max_{a \in \mathcal{A}} Q_{\pi_*}(s, a) \\
&= \max_a \mathbb{E}_{\pi_*}[r_{t+1} + \gamma R_{t+1} | s_t = s, a_t = a] \\
&= \max_a \mathbb{E}_{\pi_*}[r_{t+1} + \gamma V_*(s_{t+1}) | s_t = s, a_t = a] \\
&= \max_a \sum_{s', r} p(s', r | s, a)[r_{t+1} + \gamma V_*(s')]
\end{aligned}
$$

Equação de Bellman para Q ótimo

$$Q_*(s,a) = \mathbb{E}[r_{t+1} + \gamma \max_{a'} Q_*(s_{t+1}, a') | s_t = s, a_t = a]$$

$$Q_*(s,a) = \sum_{s',r} p(s', r|s, a)[r_{t+1} + \gamma \max_{a'} Q_*(s_{t+1}, a')]$$

Como a equação de Bellman ajuda no aprendizado se nós não temos as dinâmicas do ambiente?

Equação de Bellman

- Se não temos Q* nem p, mas temos uma estimativa de Q calculada a partir da média dos retornos experienciados, eventualmente Q->Q*

$$Q_*(s,a) = \sum_{s',r} p(s',r|s,a)[r_{t+1} + \gamma \max_{a'} Q_*(s_{t+1},a')]$$

Para atualizar uma estimativa de Q

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[R_t - Q(s_t, a_t)]$$

$$r_{t+1} + \max_{a'} Q(s', a')$$

Por hoje é “só”...

Leiam:

Richard S. Sutton and Andrew G. Barto - Reinforcement Learning: An Introduction - Second Edition

Capítulo 3