

## Inteligência Artificial

Aula 8  
 Prof<sup>a</sup> Bianca Zadrozny  
<http://www.ic.uff.br/~bianca/ia-pos>

## Tomada de decisões complexas

Capítulo 21 – Russell & Norvig  
 Seções 21.1 a 21.3

## Aprendizagem por Reforço

- Ainda temos um Processo de Decisão de Markov:
  - Um conjunto de estados  $s \in S$
  - Um conjunto de ações  $a \in A$
  - Uma modelo de transição  $T(s, a, s')$
  - Uma função de recompensa  $R(s)$
- Ainda queremos encontrar uma política  $\pi^*(s)$
- **Novidade:** o agente não conhece  $T$  e  $R$ .
  - Precisa executar ações e receber recompensas para **aprender**.

## Aprendizagem Passiva

- Tarefa Simplificada
  - O agente não conhece  $T(s, a, s')$  e  $R(s)$
  - O agente tem uma política fixa  $\pi(s)$
  - Quer aprender o quão boa é essa política = aprender a função de utilidade  $U^{\pi}(s)$ .
- Neste caso
  - Agente não escolhe as ações.
  - Apenas executa a política e aprende a partir da experiência se a política é boa ou não.
- Métodos
  - Estimativa direta da utilidade
  - Programação dinâmica adaptativa
  - Diferença temporal

## Estimativa Direta da Utilidade

- O agente executa um conjunto de episódios e observa a sequência de estados e recompensas.
 

(1, 1)<sub>0.04</sub>→(1, 2)<sub>0.04</sub>→(1, 3)<sub>0.04</sub>→(1, 2)<sub>0.04</sub>→(1, 3)<sub>0.04</sub>→(2, 3)<sub>0.04</sub>→(3, 3)<sub>0.04</sub>→(4, 3)<sub>0.1</sub>  
 (1, 1)<sub>0.04</sub>→(1, 2)<sub>0.04</sub>→(1, 3)<sub>0.04</sub>→(2, 3)<sub>0.04</sub>→(3, 3)<sub>0.04</sub>→(3, 2)<sub>0.04</sub>→(3, 3)<sub>0.04</sub>→(4, 3)<sub>0.1</sub>  
 (1, 1)<sub>0.04</sub>→(2, 1)<sub>0.04</sub>→(3, 1)<sub>0.04</sub>→(3, 2)<sub>0.04</sub>→(4, 2)<sub>0.1</sub>.
- A utilidade de um estado é calculada através da média da utilidade obtida em cada episódio a partir do estado em questão.
  - $U^{\pi}((1,1)) = (0.72 + 0.72 + (-1.16))/3 = 0.28/3$
- Converte devagar porque não utiliza as equações de Bellman.

## Programação Dinâmica Adaptativa

- Ideia:
  - Aprender o modelo de transição e a função de reforço empiricamente (ao invés das utilidades).
  - Usar iteração de valor (programação dinâmica) simplificada (sem o max) para obter as utilidades.
- Algoritmo:
  - Contar quantas vezes o estado  $s'$  ocorre quando a ação  $a$  é executada no estado  $s$ .
  - Atualizar  $R(s)$  na primeira vez em que  $s$  for visitado.

## Programação Dinâmica Adaptativa

- A partir do conjunto de episódios:

(1, 1) → (1, 2) → (1, 3) → (1, 2) → (1, 3) → (2, 3) → (3, 3) → (4, 3) +1  
 (1, 1) → (1, 2) → (1, 3) → (2, 3) → (3, 3) → (3, 2) → (3, 3) → (4, 3) +1  
 (1, 1) → (2, 1) → (3, 1) → (3, 2) → (4, 2) -1 .

pode-se calcular que

$$T((1,3), \text{Direita}, (2,3)) = 2/3$$

$$T((2,3), \text{Direita}, (3,3)) = 2/2 = 1$$

$$R((1,1)) = -0.04$$

$$R((4,3)) = +1$$

...

## Diferenças Temporais

- Ao invés de aprender modelo, aprende-se a função de utilidade diretamente, mas usando a equação de Bellman.
- A cada transição de  $s$  para  $s'$  executada, faz-se a seguinte atualização do valor de  $U^\pi(s)$ :

$$U^\pi(s) \leftarrow U^\pi(s) + \alpha(R(s) + \gamma U^\pi(s') - U^\pi(s))$$

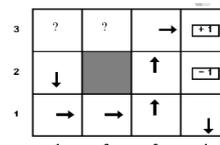
onde  $\alpha$  é a taxa de aprendizagem.

## Aprendizagem Ativa

- Agente pode escolher as ações que quiser.
- Objetivo é aprender a política ótima.
- Ideia:
  - Utilizar Programação Dinâmica Adaptativa + Iteração de Valor para encontrar política ótima usando o modelo atual.
  - Refinar modelo através da experiência e repetir.
  - Crucial: temos que dar um jeito de aprender o modelo para todos os estados e ações.

## Exemplo: Programação Dinâmica Adaptativa Gulosa

- Imagine que o agente encontra primeiro o seguinte caminho:



- Ele continuará sempre usando essa política e nunca visitará os outros estados.

## O que deu errado?

- Problema com seguir a melhor política de acordo com o modelo atual:
  - Agente não aprenderá algo sobre as melhores regiões do ambiente se a política atual nunca visita essas regiões.
- Exploração vs. Aproveitamento
  - Exploração: escolher ações sub-ótimas de acordo com modelo atual para poder melhorar o modelo.
  - Aproveitamento: escolher as melhores ações.
  - Agentes devem explorar no começo e aproveitar no final.
- Solução: escolher algumas ações aleatórias
  - Com probabilidade fixa OU
  - Com probabilidade inversamente proporcional ao número de vezes em que a ação foi executada no estado.

## Q-Learning

- Seja  $Q(a,s)$  a utilidade de se executar a ação  $a$  no estado  $s$ , isto é:

$$U(s) = \max_a Q(a, s)$$

- A função  $Q(a,s)$  é útil porque a partir dela podemos calcular diretamente a política, sem precisar de um modelo.

- Equação de atualização:

– A cada transição se  $s$  para  $s'$  quando a ação  $a$  é executada:

$$Q(a, s) \leftarrow Q(a, s) + \alpha(R(s) + \gamma \max_{a'} Q(a', s') - Q(a, s))$$