Instituto Superior de Engenharia de Coimbra

LICENCIATURA EM ENGENHARIA INFORMÁTICA

INTELIGÊNCIA COMPUTACIONAL 23-24

Aprendizagem por Reforço

---

1. Para cada um dos agentes indicados especifique o ambiente, agente, ações e recompensas:
   a. Robot aspirador.
   b. Campanha de marketing para venda de um produto com base no perfil do cliente.
   c. Controlo de temperatura de uma sala (com sistema de ar condicionado).

2. Considere um agente num mundo em grelha 3x2, conforme mostrado na figura 1. Começamos no estado "1" e terminamos no canto superior direito (estado 6). Ao atingir o estado 6, recebe uma recompensa de +10 e inicia um novo episódio. Em todas as outras ações que não levam ao estado "6", a recompensa é -1.
O agente inica na célula A e pretende alcançar F. Quando alcançar F recebe uma recompensa de +10 e o episódio termina. Para qualquer outro movimento que não conduza a F recebe a recompensa de "-1".

A start	B	C
D	E	**F finish**

Figura 1 – Ambiente.

Em cada estado temos quatro ações possíveis: cima, baixo, esquerda e direita. Para cada ação, o agente move-se de forma determinista na direção pretendida. Não são possíveis movimentos para fora da grelha.

As estimativas atuais para os valores de Q(s,a) são apresentadas na tabela abaixo:

S/A				
	Up	*Down*	*Left*	*Right*
A	**4**	**--**	**--**	**3**
B	**6**	**--**	**3**	**8**
C	**9**		**7**	
D	**--**	**2**		**5**
E	**--**	**6**	**5**	**8**

a. Considerando que existe o conhecimento completo do ambiente, atualize o valor de Q(C,left) com base na equação de Bellman, para uma política sôfrega e taxa de desconto de 0,9.

b. Assuma agora que não tem o conhecimento completo do ambiente. Atualize a Q-table com o algoritmo "SARSA". A partir de B, seguiu-se a trajetória:
"*B – baixo - E - direita – F*", terminando o episódio.
Atualize a tabela considerando coeficiente de aprendizagem de 0,2 e desconto de 0,8.

3. Considere um sistema com dois estados (S1,S2) e duas ações (a1,a2). Um agente executa ações e observa as recompensas e transições de acordo com as iterações (Estado atual; recompensa; ação; transição resultante):

It1: S1 r=-10 a1:S1->S1
It2: S1 r=-10 a2:S1->S2
It3: S2 r=+20 a1:S2->S1
It4: S1 r=-10 a2:S1->S2

a. Represente a Q-table, com entradas inicializadas a zero
b. Atualize a tabela com o algoritmo de Q-learning para as quatro iterações com coeficiente de aprendizagem de 0.5 e taxa de desconto de 0.5.

4. Como se mede o desempenho de um agente de Aprendizagem por Reforço?

Resposta:
Para medir o desempenho de um agente de aprendizagem por reforço, pode-se simplesmente somar as recompensas que recebe (normalmente com desconto).
Num ambiente simulado, executam-se muitos episódios e mede-se o valor total de recompensas que obtém em média (assim como valor mínimo, máximo, desvio padrão, etc.).

5. Na aprendizagem por reforço em que consiste o fator de desconto? A política óptima pode mudar se o fator de desconto for alterado? Justifique.

Resposta:
Ao estimar o valor de uma ação, o algoritmo de aprendizagem por reforço soma todas as recompensas a que essa ação conduz, dando mais peso às recompensas imediatas e menos peso a recompensas posteriores (considerando que uma ação tem mais influência no futuro próximo do que no futuro distante).

Para modelar este comportamento, aplica-se um fator de desconto em cada iteração (passo). Por exemplo, com um fator de desconto de 0.9, uma recompensa de 100 recebida dois passos mais tarde é contabilizada com o valor de 0,9×0,9x100 = 81 quando se estima o valor da ação.

Claramente o valor deste parâmetro tem um impacto na política ótima: se valorizarmos o futuro, podemos estar dispostos a suportar "penalizações imediatas" pela perspetiva de eventuais recompensas futuras, ao passo que se não valorizarmos o futuro, "seguramos" qualquer recompensa imediata que encontrarmos.

6. O que entende por um algoritmo de RL "off-policy"? Apresente um exemplo?

Resposta:
Um algoritmo off-line aprende o valor da política ótima (ou seja, a soma das recompensas descontadas que podem ser esperadas para cada estado se o agente agir de forma ótima), enquanto o agente executa uma política diferente. O Q-Learning é um exemplo deste tipo de algoritmo.

Em contraste, um algoritmo de política on-line aprende o valor da política que o agente efetivamente executa, incluindo tanto a política de "exploration" como a de "exploitation". O SARSA é um exemplo de um algoritmo deste tipo.

Soluções 2 e 3:

2a.

A equação de Bellman especifica indica:

$$Q(s,a) = \sum_{s'} P^a_{ss'}[R^a_{ss'} + \gamma \sum_{a'} \pi(s',a')Q(s',a')]$$

Uma vez que o ambiente é determinístico, apenas consideramos um estado seguinte s' (Probabilidade (C,left)->B =1) e sendo uma política "greedy", consideramos apenas a melhor ação a'
Q(C,left) = 1* [-1 + 0.9 * (1 * 8)] = 6.2

2b.

Para um agente SARSA - $Q(s_t, a_t) = Q(s_t, a_t) + \alpha*(r_t + \gamma*Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t))$

	SARSA policy	**Q-learning policy**
Escolher A'	ε-greedy (ε > 0 com exploration)	ε-greedy (ε > 0 com exploration)
Atualizar Q	ε-greedy (ε > 0 com exploration)	greedy policy (ε=0, sem exploration)

Assim:
Q(B, Baixo)=6+0,2*(-1+0,8*8-6)=6+0,2*(-0,6)=6-0,12=5,88
Q(E, Direita)=8 +0,2*(10+0,8*0-8)=8,4.

3.a

Q	a1	a2
S1	0	0
S2	0	0

3.b

Q-learning: $Q(state,action) \leftarrow (1-\alpha)Q(state,action)+\alpha(reward+\gamma maxaQ(next\ state,all\ actions))$

- Iteração1: S1 r=-10 a2:S1->S1

  Q(s1,a1)=(1-0,5)*0+0,5*(-10+0,5*max[0,0])=0-5=-5

Q	a1	a2
S1	-5	0
S2	0	0