Aprendizado por Reforço Livre de Modelo

Samara Ribeiro Silva
Instituto Tecnológico de Aeronáutica, Laboratório de Inteligência Artificial para Robótica Móvel (CT-213). Professor Marcus Ricardo Omena de Albuquerque Máximo, São José dos Campos, São Paulo, 05 de julho de 2021.

Na implementação de *epsilon_greedy_action* foi gerado um número aleatório e seu valor foi comparado com ε. Caso o número aleatório for maior que ε retorna-se *greedy_action* de Q, caso contrário retorna-se um número aleatório entre 0 e o tamanho de Q. Já em *greedy_action* retornou-se o índice do estado de Q com maior valor.

Para a implementação dos algoritmos *Sarsa* e *QLearning* realizou-se a seguinte atualização de $Q(S, A)$:

Atualização de $Q(S, A)$	
Sarsa	$Q(S,A) = Q(S,A) + \alpha[R + \gamma Q(S',A') - Q(S,A)]$
QLearning	$Q(S,A) = Q(S,A) + \alpha[R + \gamma max_{a' \in A}(Q(S',A')) - Q(S,A)]$

Nas figuras 1 e 2 é possível observar os resultados para o test_rl. Observe que foram obtidos resultados iguais para o *Greedy policy learnt*, mas valores distintos para a *Action-value Table*. Esses valores distintos na *Action-value Table* ocorre devido a diferença de abordagem, visto que o *Sarsa* é mais conservador que o *Qlearning* por isso tem valores maiores em módulo.

Figura 1: Resultados do test_rl para o algoritmo *Sarsa*.

```
Action-value Table:
[[ -9.42063527  -8.49171528 -10.63079058]
 [-10.50984848  -9.48866308 -11.34492207]
 [-10.98455992 -10.46413289 -11.34822116]
 [-11.69885225 -11.31849468 -12.08481955]
 [-12.28266751 -12.2246947  -12.24279906]
 [-11.5590788  -12.05730971 -11.37286469]
 [-11.13595356 -11.60674814 -10.404567  ]
 [-10.28641454 -11.35397048  -9.36183345]
 [ -9.35982048 -10.37946269  -8.47529658]
 [ -7.31527013  -8.45361902  -8.50487916]]
Greedy policy learnt:
[L, L, L, L, L, R, R, R, R, S]
```

Figura 2: Resultados do test_rl para o algoritmo *QLearning*.

```
Action-value Table:
[[-1.99       -1.         -2.9701    ]
 [-2.96880737 -1.99       -3.92508565]
 [-3.70115172 -2.9701     -4.38982482]
 [-4.43453348 -3.94039892 -4.72290021]
 [-5.115267   -4.89001033 -4.89094129]
 [-4.13815667 -4.68609652 -3.94039883]
 [-3.60784076 -4.37847851 -2.9701    ]
 [-2.96740325 -3.92750133 -1.99      ]
 [-1.99       -2.9701     -1.        ]
 [ 0.         -0.99       -0.99      ]]
Greedy policy learnt:
[L, L, L, L, L, R, R, R, R, S]
```

Nas figuras 3 a 10 pode-se observar os resultados obtidos. Observe que a tabela de *Greedy policy* é bem semelhante e os valores na *Action-value Table* do Sarsa se mantiveram maiores em

módulo. A convergência do QLearning é mais rápida que o Sarsa devido ao comportamento menos conservador se comparado com o Sarsa.

Figura 3: Recompensa acumulada para o algoritmo *Sarsa*.

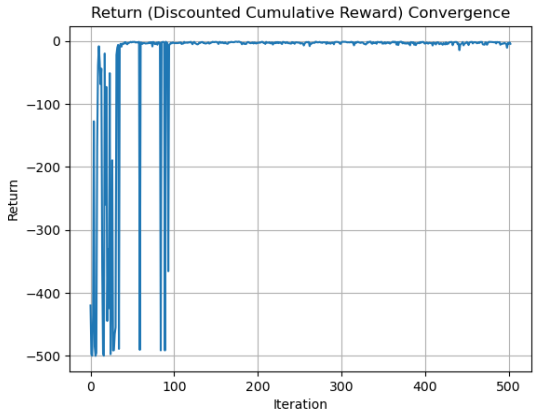

Figura 4: Percurso para o algoritmo *Sarsa*.

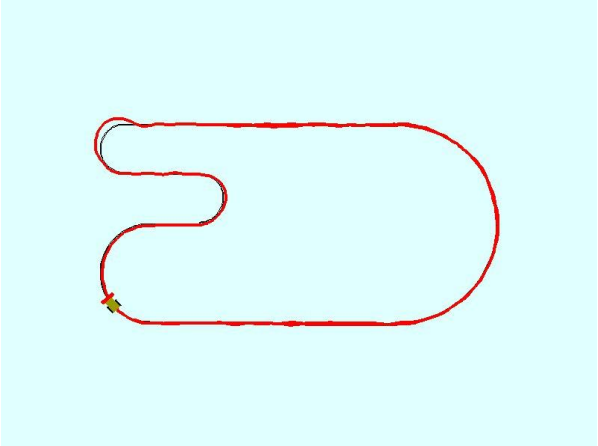

Figura 5: Tabela Greedy Police para o algoritmo *Sarsa*.

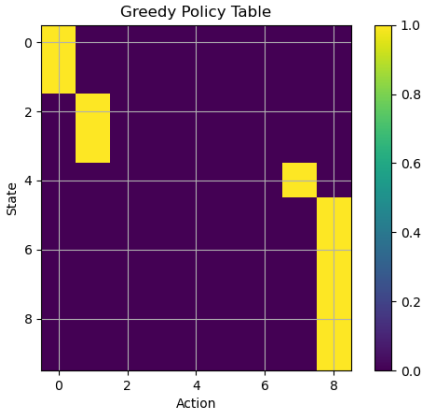

Figura 6: Tabela Action-Value para o algoritmo *Sarsa*.

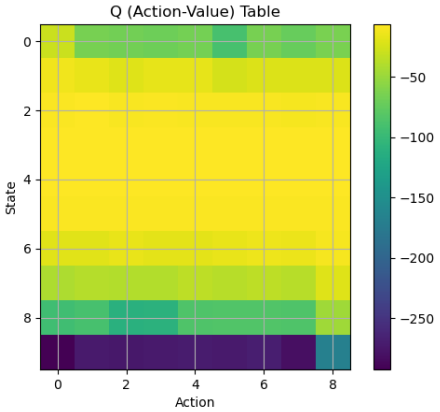

Figura 7: Recompensa acumulada para o algoritmo *QLearning*.

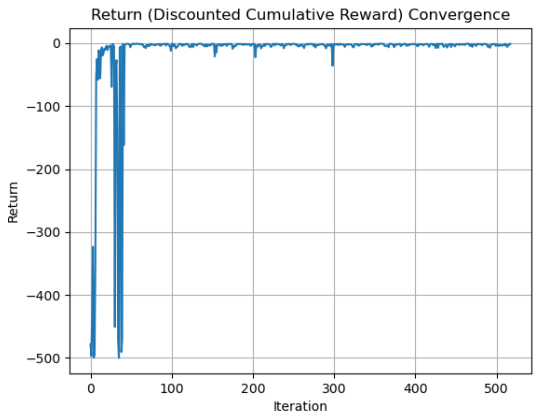

Figura 8: Percurso para o algoritmo *QLearning*.

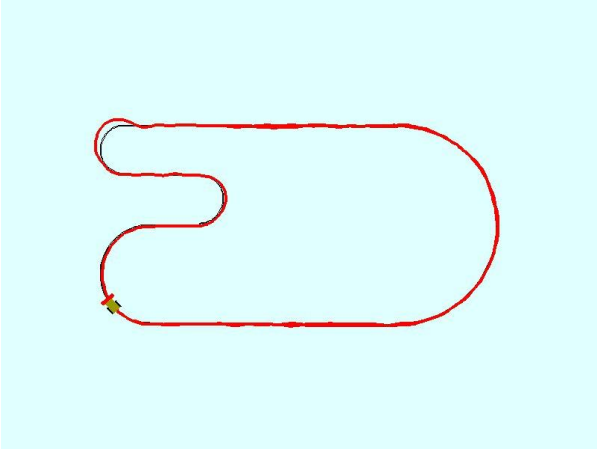

Figura 9: Tabela Greedy Police para o algoritmo *QLearning*.

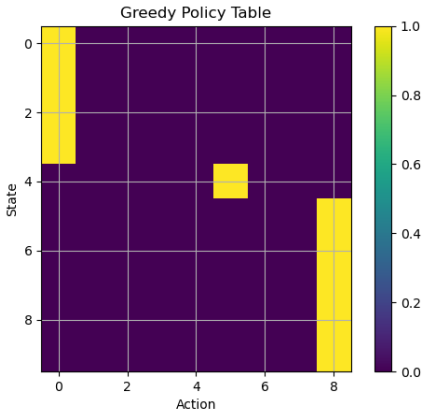

Figura 10: Tabela Action-Value para o algoritmo *QLearning*.