Problema de navegação em ambiente com deslocamentos involuntários

Izabella Thaís Oliveira Gomes
12/0152401
igomesizabella@gmail.com

Letícia Câmara van der Ploeg
12/0152771
leticiacvdploeg@gmail.com

Junho/2016

Resumo

Este documento relata a resolução de um problema de aprendizado por reforço que consiste em uma navegação em ambiente com deslocamentos involuntários utilizando os algoritmos Sarsa e Q-learning. O agente move-se podendo ser deslocado pelo vento em determinados espaços.

I. Introdução

O aprendizado por reforço está presente na natureza, no dia-a-dia e é bastante intuitivo para os seres humanos. A ideia de interagir com o ambiente configura uma maneira de aprendizado na qual nós aprendemos sozinhos, realizando ações e observando suas consequências positivas ou negativas e guardando essa experiência para que no futuro possamos realizar as melhores escolhas que nos levam ao objetivo final. A abordagem computacional de aprendizagem por reforço é uma das técnicas de inteligência artificial que pressupõe ambientes desconhecidos, ausência de supervisão e a relação de causa e efeito decorrente das ações tomadas. Os diferentes algoritmos de aprendizado por reforço, como Monte Carlo, SARSA e Q-learning, são efetivos em resolver problemas de natureza científica ou econômica, utilizando experiência computacional ou análise matemática, com foco no objetivo e nas recompensas ou punições dadas.

A cada instante de tempo o agente está em um estado **s**, executa uma ação **a**, que direciona para um estado **s'** e recebe uma recompensa **r**. Objetivamente, um algoritmo de aprendizado por reforço procura uma solução baseado na escolha de uma política π que maximize as recompensas obtidas pelo agente. A política π mapeia estados em ações. Para saber se um estado é "bom"ou "ruim"existe a cada estado uma função de valor associada dada por V_π que representa a recompensa de um estado s somada com recompensas futuras se seguir uma política de ações π. Pode estar associado a função de valor um fator de desconto que garante convergência e diferencia recompensas distantes do estado atual. Analogamente, tem-se a função de valor de ações, dada por V_π, na qual o valor da ação é a recompensa da ação somada com o valor do estado para onde o agente vai devido à ação. Como aprender uma política ótima designa os algoritmos de aprendizado por reforço já citados.

i. Algoritmo Q-learning

Este é o algoritmo de aprendizado por reforço mais utilizado e estabelece autonomamente

uma política de ações interativamente.

Pode-se demonstrar que o algoritmo Q-learning converge para um procedimento de controle ótimo, quando a hipótese de aprendizagem de paresr representada por uma estado-ação Q fo tabela completa contendo a informação de valor de cada par. A convergência ocorre tanto em processos de decisão Markovianos determinísticos quanto não-determinísticos [3]. O algoritmo é mostrado na figura.

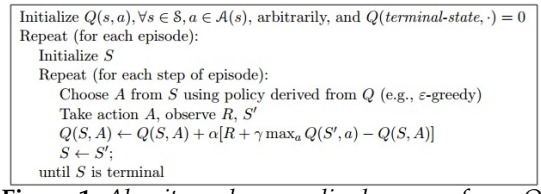
Initialize $Q(s,a), \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
  Initialize S
  Repeat (for each step of episode):
    Choose A from S using policy derived from Q (e.g., ε-greedy)
    Take action A, observe R, S'
    $Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma \max_a Q(S',a) - Q(S,A)]$
    $S \leftarrow S'$;
  until S is terminal

Figura 1: *Algoritmo de aprendizado por reforço Q-learning [1]*

ii. Algoritmo Sarsa

O algoritmo Sarsa converge para uma política e valor de função de ação ótimos assim que todos os pares estado-ação tenham sido visitados um número infinito de vezes e a política de escolha da próxima ação convirja, no limite, para uma política que utilize a melhor ação (ou seja, aquela que maximize a recompensa futura esperada) [3]. O algoritmo é mostrado na figura 2.

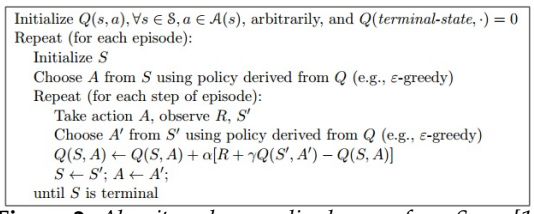
Initialize $Q(s,a), \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
  Initialize S
  Choose A from S using policy derived from Q (e.g., ε-greedy)
  Repeat (for each step of episode):
    Take action A, observe R, S'
    Choose A' from S' using policy derived from Q (e.g., ε-greedy)
    $Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma Q(S',A') - Q(S,A)]$
    $S \leftarrow S'$; $A \leftarrow A'$;
  until S is terminal

Figura 2: *Algoritmo de aprendizado por reforço Sarsa [1]*

II. Metodologia

O problema envolve um espaço de configurações dado por uma grade com estados inicial e final (objetivo), onde todos os estados podem ocasionar deslocamentos involuntários. O padrão de movimentos envolve deslocamentos para cima, baixo, esquerda e direita, ou seja, 4 ações possíveis. Já os deslocamentos involuntários para cima são decorrentes de uma frente de vento que está associada a cada coluna da grade, com intensidade 0,1 ou 2, como mostrado na figura 3. A tarefa não possui descontos, para cada movimento tem-se uma recompensa de -1 (punição) até que o estado objetivado seja atingido e aquele movimento que leva o agente a sair da grade não implica em punição adicional e não altera a localização atual do agente. A solução do problema é feita utilizando aprendizado por reforço.

A implementação realizada apresenta 5 arquivos de códigos no Matlab:

navigation_problem.m: Definição do espaço de configurações e seleção de algoritmo de aprendizado por reforço a ser utilizado;

sarsa_algorithm.m: Implementação do método DT (diferença temporal) Sarsa, que retorna os episódios, timesteps, a função de valor de ação (Q) e a sequência de passos realizados até atingir o objetivo;

next_state_and_reward.m: Função que calcula a recompensa e próximo estado dada uma ação, considerando a frente de vento;

plot_policy_in_gridworld.m: Função para plotar resultados;

q_learning_algorithm.m: Implementação do método DT (diferença temporal) Q-learning, que retorna episódios, timesteps, a função de valor de ação (Q) e a sequência de passos realizados até atingir o objetivo.

As seções seguintes detalham a solução do problema ao apresentar e analisar os resultados dos algoritmos para 10000 episódios.

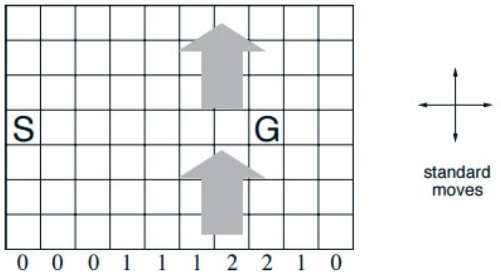

Figura 3: *Espaço de configurações do problema, onde cada movimento é influenciado por uma frente de vento*

III. Resultados e análise

O espaço de configurações é mostrado na figura 4, que indica a presença da frente de vento. A cores azul, violeta e rosa apresentam, respectivamente, intensidades nula, 1 e 2 de frente de vento. Os pontos inicial e final também são mostrados.

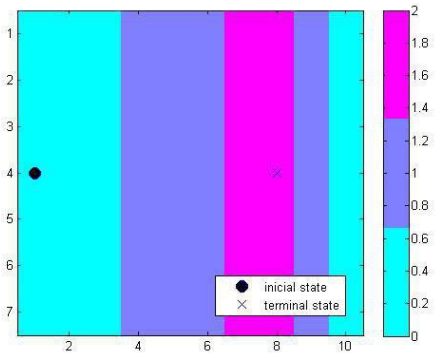

Figura 4: *Grade com pontos inicial e final e indicação de regiões afetadas pela frente de vento e sua intensidade*

Com os algoritmos de aprendizado por reforço **Sarsa** e **Q-learning** a solução do problema se inicia com a definição da função de valor de ação inicial, a figura 5 apresenta o espaço de configurações com a função de valor de estado para cada estado, que é obtida pela fórmula:

$$V_t(s_t + 1) = maxQ(s_t + 1, a) \qquad (1)$$

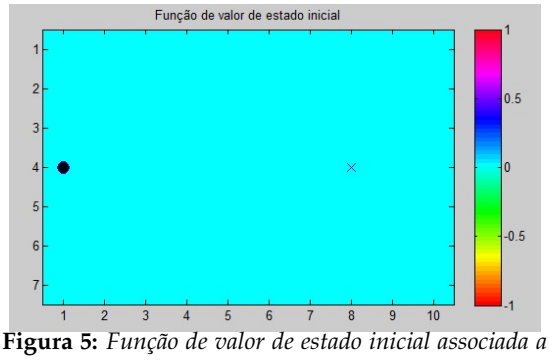

Figura 5: *Função de valor de estado inicial associada a cada estado da grade*

Como explicitam os algoritmos, o valor inicial de V é nulo para todos os estados. Ao executar o Sarsa e realizar o aprendizado por reforço, obtém-se o resultado apresentado na figura 6. Para a execução do Q-learning tem-se o resultado apresentado na figura 7.

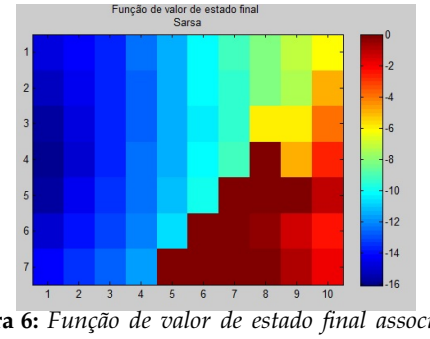

Figura 6: *Função de valor de estado final associada a cada estado da grade para Sarsa*

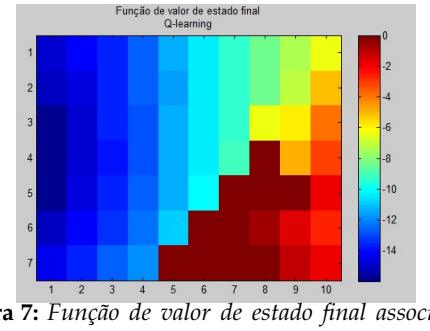

Figura 7: *Função de valor de estado final associada a cada estado da grade para Q-learning*

Perceba que, mesmo após o aprendizado, existem estados com valor nulo (cor vinho). Estes

estados possuem valor nulo porque nunca são atingidos devido a frente de vento presente na região destes. As figuras 8 e 9 mostram os resultados ao longo do aprendizado baseado nos passos necessários para cada episódio, ou seja, o número de passos executados para sair do estado inicial e chegar ao estado final.

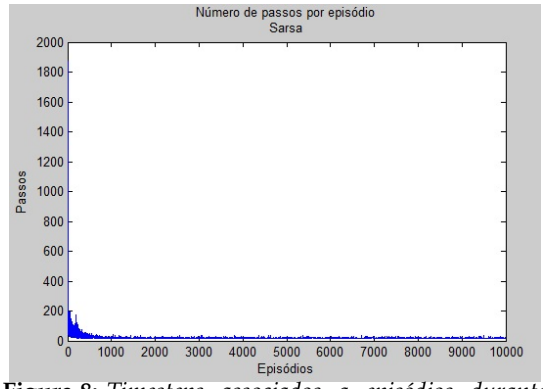

Figura 8: *Timesteps associados a episódios durante aprendizado*

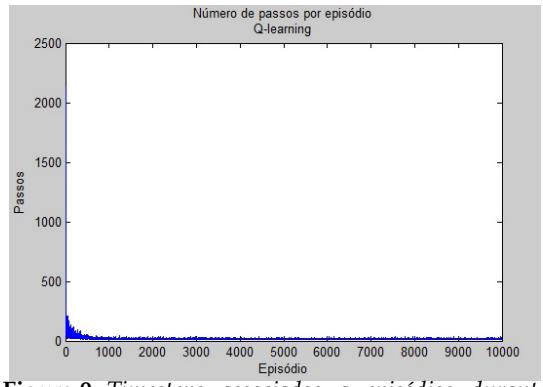

Figura 9: *Timesteps associados a episódios durante aprendizado*

Perceba que no início do aprendizado um episódio dura muito mais que os próximos. A curva converge para um valor ótimo. O algoritmo Sarsa encontra sua política ótima em *16 passos*, enquanto que o Q-learning encontra em *17 passos*. Desta forma o método Sarsa se mostra mais eficiente. As figuras 10 e 11 mostram a política ótima encontrada pelos algoritmos Sarsa e Q-learning.

Os caminhos encontrados ao final dos 10000 episódios para os algoritmos Sarsa e Q-learning são mostrados nas figuras 12 e 13.

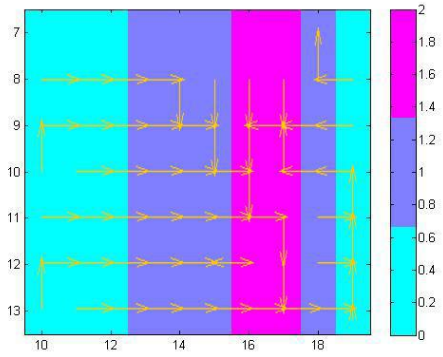

Figura 10: *Política ótima encontrada pelo algoritmo Sarsa*

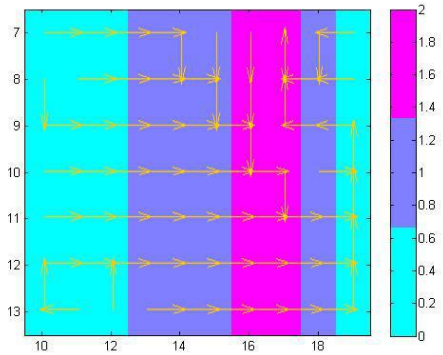

Figura 11: *Política ótima encontrada pelo algoritmo Q-learning*

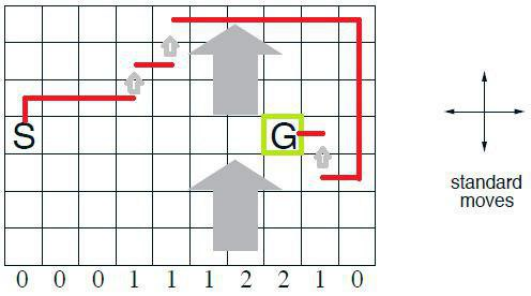

Figura 12: *Caminho ótimo encontrado pelo algoritmo Sarsa ao fim dos 10000 episódios*

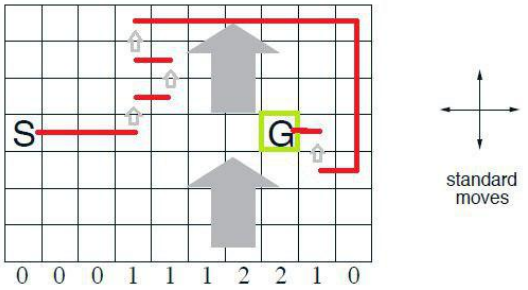

Figura 13: *Caminho ótimo encontrado pelo algoritmo Q-learning ao fim dos 10000 episódios*

IV. Conclusão

O aprendizado por reforço foi aplicado na resolução do problema de navegação com deslocamentos involuntários utilizando os algoritmos SARSA e Q-learning. Os algoritmos são muito semelhantes e diferem apenas na forma como a função de valor de ação é atualizada. Os dois algoritmos foram treinados por 10000 episódios encontrando uma política e um caminho ótimo. O Sarsa demonstrou desempenho levemente superior ao Q-learning, uma vez que saiu do ponto inicial ao ponto final do espaço de configurações em 16 passos, enquanto o outro executou 17 passos. A dinâmica associada a convergência, por sua vez, é bastante semelhante para os dois algoritmos.

Referências

[1] Richard S. Sutton and Andrew G. Barto. Reinforcement Learning: An Introduction, 2012. MIT Press.

[2] Ronaldo Prati. Notas de aula: Inteligência Artificial, UFABC. Disponível em: <http://professor.ufabc.edu.br/ ronaldo.prati/InteligenciaArtificial/ reinforcement-learning.pdf>

[3] Sildomar T. Monteiro; Carlos H. C. Ribeiro. Desempenho de algoritmos de aprendizagem por reforço sob condições de ambiguidade sensorial em robótica móvel. Disponível em: <http://www.scielo.br/scielo.php?script= sci_arttext&pid=S0103-17592004000300008>