Instituto Tecnológico de Aeronáutica

CT-213

Laboratório 10 - Aprendizado por Reforço com Programação Dinâmica

Aluno: Pedro Elardenberg Sousa e Souza
Professor: Marcos Ricardo Omena de Albuquerque Maximo

30 de abril de 2024

Conteúdo

1 Resumo

Neste Laboratório, foram implementados algoritmos de programação dinâmica no contexto de solução de um Processo Decisório de Markov *(Markov Decision Process - MDP)*. Os algoritmos implementados foram avaliação de política *(policy evaluation)*, iteração de política *(policy iteration)* e iteração de valor *(value iteration)*.

O objetivo desses algoritmos neste contexto foi de avaliar políticas e determinar políticas ótimas para um *grid world*. Os algoritmos desenvolvidos resolvem o problema de Aprendizado por Reforço *(Reinforcement Learning - RL)* no caso em que o MDP é conhecido.

Os resultados foram então mostrados e discutidos neste relatório.

2 Introdução

2.1 Equação de Bellman

Os algoritmos desenvolvidos neste laboratório seguem-se dos resultados obtidos a partir da Equação de Bellman. Essa equação resolve o problema da tomada de decisão ótima no contexto de *Reinforcement Learning*, e também em outras áreas da ciência [1].

Quando se possui um problema se pode assumir que ele se comporta como um ambiente de Markov, ou seja, o estado seguinte do meu sistema depende apenas de seu estado inicial e a ação tomada naquele estado, não importando os processos anteriores, pode-se utilizar da Equação de Bellman de Expectativa para prever qual a função estado-valor v_π para cada um dos estados s ou qual função ação-valor q_π para cada estado s e ação a dado um desconto γ.

$$\begin{aligned} v_\pi(s) &= E_\pi[R_{t+1} + \gamma R_{t+2} + \cdots | S_t = s] \\ v_\pi(s) &= E_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s] \\ q_\pi(s,a) &= E_\pi[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) | S_t = s, A_t = a] \end{aligned} \tag{1}$$

Em que E é a esperança ou valor esperado da expressão entre colchetes.

Uma função estado-valor v_π pode ser escrita como se fosse uma combinação linear entre todas as ações que se pode tomar, dada uma política $\pi(a|s)$ vezes a função ação-valor q_π associada a cada ação dado que se está no estado s. Ou seja:

$$v_\pi(s) = \sum_{a \in A} \pi(a|s) q_\pi(s,a) \tag{2}$$

O valor de $q_\pi(s,a)$ na equação 1, aplicando-se a esperança nos termos da expressão, obtém-se:

$$q_\pi(s,a) = r(s,a) + \gamma \sum_{s' \in S} p(s'|s,a) v_\pi(s') \tag{3}$$

Em que $r(s,a)$ é a função de recompensa e $E_\pi[\gamma q_\pi(S_{t+1}, A_{t+1}) | S_t = s, A_t = a]$ equivale a tomar cada ação no estado s' que da função de dinâmica $p(s'|s,a)$ que se pode chegar partindo de s vezes a função-valor v_π desses estados s'. Juntando as equações 2 e 3, obtém-se a Equação de Bellman de Expectativa:

$$v_\pi(s) = \sum_{a \in A} \pi(a|s) + \left(r(s,a) + \gamma \sum_{s' \in S} p(s'|s,a) v_\pi(s') \right) \tag{4}$$

Para se encontrar a política ótima, utiliza-se a Equação de Bellman de Otimalidade. Utilizando as equações 3 e 4 para encontrar a função valor ótima $v_*(s)$ e função ação-valor ótima $q_*(s,a)$:

$$\begin{aligned} v_*(s) &= \max_a \left(r(s,a) + \gamma \sum_{s' \in S} p(s'|s,a) v_*(s') \right) \\ q_*(s,a) &= r(s,a) + \gamma \sum_{s' \in S} p(s'|s,a) \max_{a' \in A} q_*(s',a') \end{aligned} \tag{5}$$

2.2 Avaliação de Política

Juntando as equações 2 e 3, otém-se:

$$v_\pi(s) = \sum_{a \in A} r(s,a) + \gamma \sum_{a \in A} \sum_{s' \in S} \pi(a|s) p(s'|s,a) v_\pi(s') \tag{6}$$

Como o cálculo é feito sobre todos os estados s possíveis, pode-se usar o conceito de Programação Dinâmica para se calcular a função-valor que dá a avaliação da política do meu sistema. Reescrevendo, portanto, a equação 6:

$$v_{k+1}(s) = \sum_{a \in A} r(s,a) + \gamma \sum_{a \in A} \sum_{s' \in S} \pi(a|s) p(s'|s,a) v_k(s') \tag{7}$$

Partindo-se de uma função-valor qualquer, o sistema converge para a função-valor da política $v_\pi(s)$.

2.3 Iteração de Valor

Com a mesma ideia, pode-se reescrever a Equação de Bellman de Otimalidade 5 nos termos da equação 7:

$$v_{k+1}(s) = \max_a \left(r(s,a) + \gamma \sum_{s' \in S} p(s'|s,a) v_k(s') \right) \tag{8}$$

Que converge para a função-valor da política ótima $v_*(s)$.

Por fim, quando o algoritmo tiver convergido, encontra-se a política ótima:

$$\pi_*(s) = greedy(v_*(s)) \tag{9}$$

2.4 Iteração de Política

Ainda com a mesma ideia de usar Programação Dinâmica, para se encontrar a política ótima pode-se inicar com política e função-valor aleatórias e avaliar a política usando avaliação de política iterativa (equação 7) e em seguida melhorar a política agindo de forma gulosa em relação a $v_k(s)$.

Não é necessário calcular a função-valor ótima para cada política recalculada. Basta fazer o processo de cálculo descrito em 7 e aplicar a solução gulosa dessa solução a partir dele.

$$\pi'(s) = greedy(v_\pi(s)) \tag{10}$$

3 Análise dos Resultados

Os algoritmos de *Reinforcement Learning* implementados servem para resolver o problema de avaliar políticas e determinar políticas ótimas para um *grid world*, conforme ilustra a figura 1. esses algoritmos resolvem o problema de Aprendizado por Reforço no caso em que o modelo do Processo Decisório de Markov (MDP) é conhecido.

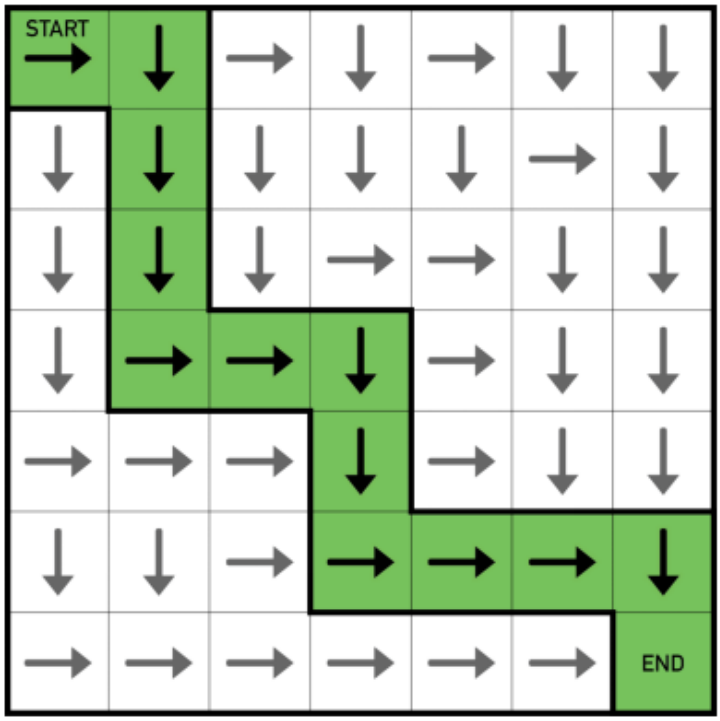

Figura 1: exemplo de política ótima em *grid world*

Os algoritmos implementados neste laboratório utilizam os conceitos de programação dinâmica e as fórmulas apresentadas na seção anterior. Assim, além das funções criadas para encontrar Avaliação de Política, Iteração de Valor e Iteração de Política, criou-se uma função auxiliar, chamada *value_pd*, que fazia os cálculos comuns a todos os algoritmos, ou seja, os valores dados pelas equações 2 e 3.

3.1 Avaliação de Política

Utilizando a equação 7, e utilizando uma política de escolha aleatória para cada estado, o método *policy_evaluation* retorna a seguinte avaliação:

```
Evaluating random policy, except for the goal state, where policy always executes stop:
Value function:
[  -384.09,  -382.73,  -381.19,    *    ,  -339.93,  -339.93]
[  -380.45,  -377.91,  -374.65,    *    ,  -334.92,  -334.93]
[  -374.34,  -368.82,  -359.85,  -344.88,  -324.92,  -324.93]
[  -368.76,  -358.18,  -346.03,    *    ,  -289.95,  -309.94]
[    *    ,  -344.12,  -315.05,  -250.02,  -229.99,    *    ]
[  -359.12,  -354.12,    *    ,  -200.01,  -145.00,     0.00]
Policy:
[  SURDL  ,  SURDL  ,  SURDL  ,    *    ,  SURDL  ,  SURDL  ]
[  SURDL  ,  SURDL  ,  SURDL  ,    *    ,  SURDL  ,  SURDL  ]
[  SURDL  ,  SURDL  ,  SURDL  ,  SURDL  ,  SURDL  ,  SURDL  ]
[  SURDL  ,  SURDL  ,  SURDL  ,    *    ,  SURDL  ,  SURDL  ]
[    *    ,  SURDL  ,  SURDL  ,  SURDL  ,  SURDL  ,    *    ]
[  SURDL  ,  SURDL  ,    *    ,  SURDL  ,  SURDL  ,    S    ]
-----------------------------------------------------------------
```

Figura 2: Avaliação de política aleatória para $\gamma = 1.0$ e probabilidade de escolha correta da ação = 1.0

3.2 Iteração de Valor

Implementando o algoritmo de cálculo da equação 8, obteve-se o seguinte resultado:

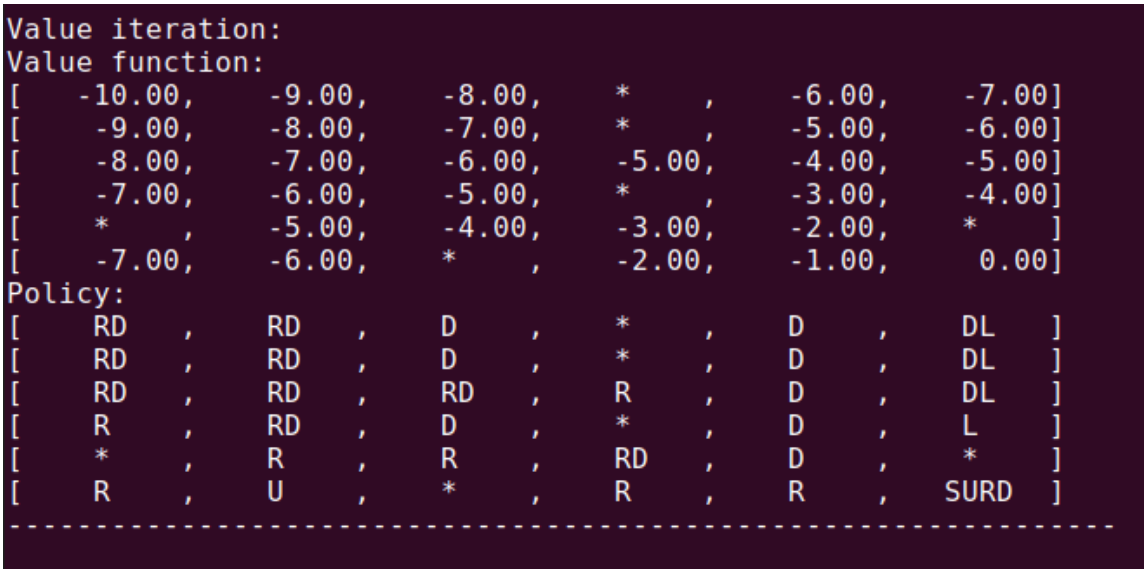

Figura 3: Iteração de valor para $\gamma = 1.0$ e probabilidade de escolha correta da ação = 1.0

Observa-se que o algoritmo encontrou a solução ótima, que é dada pela distância, para cada célula, da célula do canto inferior direito, considerando os obstáculos assinalados em ***** do *grid world*.

3.3 Iteração de Política

Implementando o algoritmo de iteração política, utilizando as equações 7 e 10, obtém-se a política ótima e a função-valor ótima a ela associada, partindo de valores inicias nulos para cada função, como se vê na figura 4

```
Policy iteration:
Value function:
[   -10.00,    -9.00,    -8.00,     *    ,    -6.00,    -7.00]
[    -9.00,    -8.00,    -7.00,     *    ,    -5.00,    -6.00]
[    -8.00,    -7.00,    -6.00,    -5.00,    -4.00,    -5.00]
[    -7.00,    -6.00,    -5.00,     *    ,    -3.00,    -4.00]
[     *    ,    -5.00,    -4.00,    -3.00,    -2.00,     *   ]
[    -7.00,    -6.00,     *    ,    -2.00,    -1.00,     0.00]
Policy:
[    RD   ,    RD   ,    D    ,    *    ,    D    ,    DL   ]
[    RD   ,    RD   ,    D    ,    *    ,    D    ,    DL   ]
[    RD   ,    RD   ,    RD   ,    R    ,    D    ,    DL   ]
[    R    ,    RD   ,    D    ,    *    ,    D    ,    L    ]
[    *    ,    R    ,    R    ,    RD   ,    D    ,    *    ]
[    R    ,    U    ,    *    ,    R    ,    R    ,   SURD  ]
------------------------------------------------------------
```

Figura 4: Iteração de política para $\gamma = 1.0$ e probabilidade de escolha correta da ação = 1.0

3.4 Comparação entre Grid Worlds diferentes

Por fim, utilizou-se um *grid world* com o desconto $\gamma = 0.98$ e probabilidade de escolha correta da ação = 0.8, sendo aplicados os mesmos algoritmos. Os resultados obtidos são mostrados a seguir:

```
Evaluating random policy, except for the goal state, where policy always executes stop:
Value function:
[   -47.19,   -47.11,   -47.01,    *    ,   -45.13,   -45.15]
[   -46.97,   -46.81,   -46.60,    *    ,   -44.58,   -44.65]
[   -46.58,   -46.21,   -45.62,   -44.79,   -43.40,   -43.63]
[   -46.20,   -45.41,   -44.42,    *    ,   -39.87,   -42.17]
[    *    ,   -44.31,   -41.64,   -35.28,   -32.96,    *   ]
[   -45.73,   -45.28,    *    ,   -29.68,   -21.88,     0.00]
Policy:
[  SURDL  ,  SURDL  ,  SURDL  ,    *    ,  SURDL  ,  SURDL  ]
[  SURDL  ,  SURDL  ,  SURDL  ,    *    ,  SURDL  ,  SURDL  ]
[  SURDL  ,  SURDL  ,  SURDL  ,  SURDL  ,  SURDL  ,  SURDL  ]
[  SURDL  ,  SURDL  ,  SURDL  ,    *    ,  SURDL  ,  SURDL  ]
[    *    ,  SURDL  ,  SURDL  ,  SURDL  ,  SURDL  ,    *   ]
[  SURDL  ,  SURDL  ,    *    ,  SURDL  ,  SURDL  ,    S   ]
------------------------------------------------------------
```

Figura 5: Avaliação de política aleatória para $\gamma = 0.98$ e probabilidade de escolha correta da ação = 0.8

Percebe-se que a função-valor nesse caso ficou menor do que na configuração anterior. Isso se dá majoritariamente por causa do desconto γ, pois, no cálculo desta avaliação de política, a política usada é a de se tomar qual-

quer direção possível aleatoriamente em cada estado. Isso cria, naturalmente, alguns caminhos muito grandes, que não são atenuados pelo fator desconto. Assim, ao se calcular a esperança do retorno, a função dá um valor maior. Testando valores diferentes para a probabilidade de escolha correta da ação, vê-se que as penalidades ficam ligeiramente maiores, mas que são de menos influência que o valor do desconto.

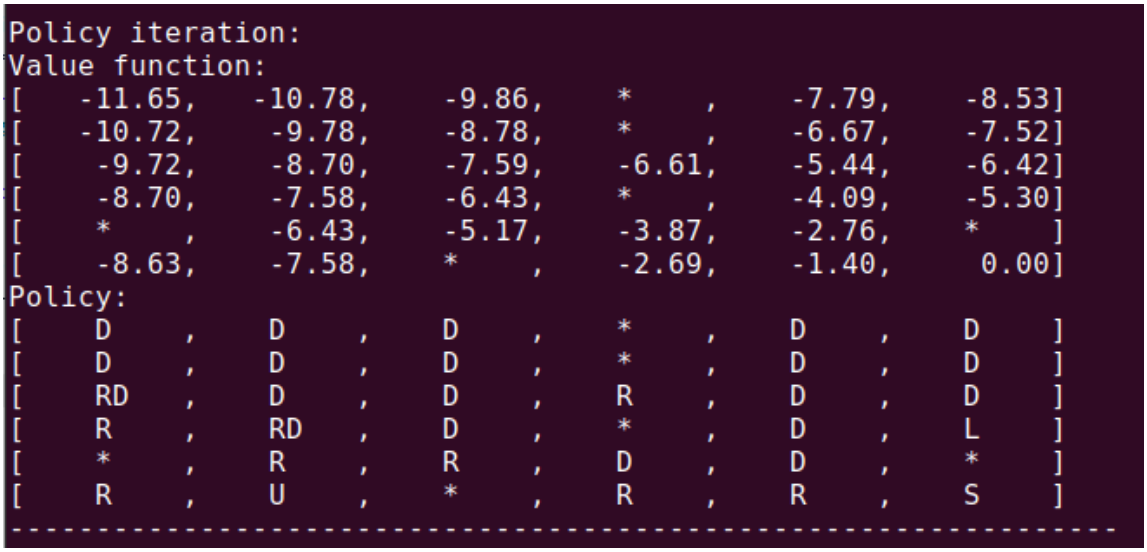

Figura 6: Iteração de valor para $\gamma = 0.98$ e probabilidade de escolha correta da ação $= 0.8$

Neste caso, como a política é ótima, o desconto γ não é fator determinante na mudança dos valores. Assim, para uma probabilidade de escolha correta da ação menor, o algoritmo encontra caminhos um pouco mais longos em relação à situação anterior, obtendo-se penalidades maiores.

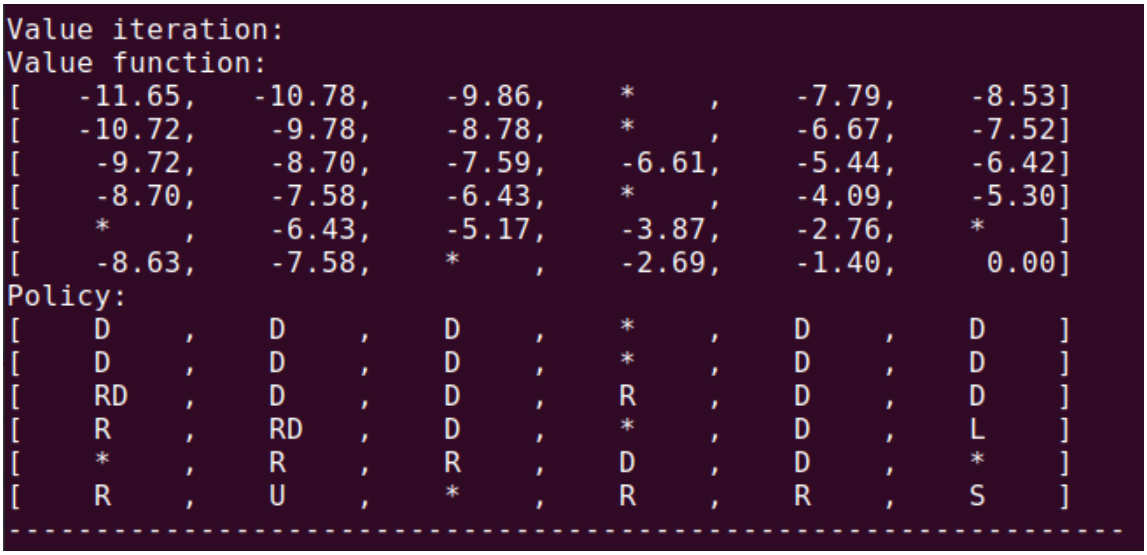

Figura 7: Iteração de política para $\gamma = 0.98$ e probabilidade de escolha correta da ação = 0.8

Para a iteração de política, o resultado é similar ao de iteração de valor, encontrando uma política "ótima"igual à da figura 6.

4 Conclusão

O presente laboratório mostrou que algoritmos de *Reinforcement Learning* são muito eficazes na avaliação de problemas que se encaixam num contexto em que o modelo de Processo Decisório de Markov é conhecido. Os algoritmos puderam ser treinados de forma a avaliar corretamente uma política, encontrar a função-valor ótima para cada estado e encontrar a política ótima para o problema dado.

A execução dos mesmos algoritmos com valores de desconto e probabilidade de escolha correta da ação diferentes mostra que o sistema consegue convergir para valores próximos do ótimo mesmo quando a probabilidade da escolha correta não é 1.0, dando a entender que, para sistemas em que essa probabilidade não pode ser definida, isto é, a incerteza é própria do problema, o problema ainda vai convergir para valores próximos. Um desconto menor que 1.0 ajuda ao modelo obter melhores resultados devido à desconsideração no longo prazo de caminhos muito grandes.

Referências

[1] Marcos Ricardo Omena de Albuquerque Maximo. Ct-213 - aula 11 - introdução ao aprendizado por reforço. Apostila, 2020. Curso CT-213 - Inteligência Artificial para Robótica Móvel, Instituto Tecnológico de Aeronáutica.