Instituto Tecnológico de Aeronáutica - ITA
Inteligência Artificial para Robótica Móvel - CT-213
Aluno: Danilo de Farias Matos

Relatório do Laboratório 10 - Programação Dinâmica

1. Breve Explicação em Alto Nível da Implementação

1.1. Avaliação de Política

1.1.1. Iniciei a implementação da avaliação de política com um loop “for” para iterar sobre as 10.000 iterações sugeridas, dentro deste loop declarei uma variável delta para comparar com epsilon e verificar se a mudança estava inferior (caracterizando convergência) e quebrar o loop antecipadamente. Após isso, copiei os valores passados para conseguir iterar de maneira síncrona os valores da política e coloquei mais um loop “for” para iterar sobre todos os estados e obter o valor daquele estado de acordo com a política a ser avaliada e ao fim da iteração atualizar os valores para a nova interação completa sobre os estados. Esta função não pensa em buscar a política ótima, apenas utiliza uma dada política e atualiza os valores obtidos em cada um dos estados após repetir n vezes o uso da política.

1.2. Iteração de Valor

1.2.1. Para abordagem da iteração do valor, inicializei o algoritmo de maneira muito similar ao da avaliação de política. A diferença agora está em utilizar uma variável (max_action_value) para atualizar o maior valor possível encontrado “q(s,a)” após as iterações para cada estado. Nesta implementação não nos importamos com uma política específica, apenas iteramos e atualizamos os valores com base nas maiores recompensas até encontrarmos um caminho ótimo.

1.3. Iteração de Política

1.3.1. Para implementar a iteração de políticas eu criei um loop for para iterar sobre as quantidades de iterações propostas (10000) e utilizei o valor atualizado dos estados com base em 3 iterações da função “policy_evaluation()” após isso eu copiei a policy inicial para comparar com a nova policy (obtida por meio da função “gredy_policy()”). Para atualizar a política com a função greedy eu usei o valor obtido por meio das 3 iterações sobre a política inicial. Por fim coloquei um early stop que é ativado se a política nova for igual à anterior.

2. Tabelas Comprovando Funcionamento do Código

Basta colocar os *prints* das tabelas (ou cópias das saídas do programa)

2.1. Caso $p_c = 1,0$ e $\gamma = 1,0$

2.1.1. Avaliação de Política

```
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    ]
------------------------------------------------------------
```

2.1.2. Iteração de Valor

```
Value 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  ]
------------------------------------------------------------------
```

2.1.3. Iteração de Política

```
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  ]
----------------------------------------------------------------
```

2.2. Caso $p_c = 0,8$ e $\gamma = 0,98$

2.2.1. Avaliação de Política

```
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    ]
----------------------------------------------------------------
```

2.2.2. Iteração de Valor

```
Value iteration:
Value function:
[   -11.65,   -10.78,    -9.86,    *    ,    -7.79,    -8.53]
[   -10.72,    -9.78,    -8.78,    *    ,    -6.67,    -7.52]
[    -9.72,    -8.70,    -7.59,    -6.61,    -5.44,    -6.42]
[    -8.70,    -7.58,    -6.43,    *    ,    -4.09,    -5.30]
[    *    ,    -6.43,    -5.17,    -3.87,    -2.76,    *    ]
[    -8.63,    -7.58,    *    ,    -2.69,    -1.40,     0.00]
Policy:
[    D    ,    D    ,    D    ,    *    ,    D    ,    D    ]
[    D    ,    D    ,    D    ,    *    ,    D    ,    D    ]
[    RD   ,    D    ,    D    ,    R    ,    D    ,    D    ]
[    R    ,    RD   ,    D    ,    *    ,    D    ,    L    ]
[    *    ,    R    ,    R    ,    D    ,    D    ,    *    ]
[    R    ,    U    ,    *    ,    R    ,    R    ,    S    ]
------------------------------------------------------------
```

2.2.3. Iteração de Política

```
Policy iteration:
Value function:
[   -11.66,   -10.79,    -9.87,    *    ,    -7.80,    -8.53]
[   -10.73,    -9.79,    -8.78,    *    ,    -6.67,    -7.52]
[    -9.73,    -8.70,    -7.60,    -6.61,    -5.44,    -6.42]
[    -8.70,    -7.58,    -6.43,    *    ,    -4.09,    -5.30]
[    *    ,    -6.43,    -5.18,    -3.87,    -2.76,    *    ]
[    -8.64,    -7.58,    *    ,    -2.69,    -1.40,     0.00]
Policy:
[    D    ,    D    ,    D    ,    *    ,    D    ,    D    ]
[    D    ,    D    ,    D    ,    *    ,    D    ,    D    ]
[    R    ,    D    ,    D    ,    R    ,    D    ,    D    ]
[    R    ,    D    ,    D    ,    *    ,    D    ,    L    ]
[    *    ,    R    ,    R    ,    D    ,    D    ,    *    ]
[    R    ,    U    ,    *    ,    R    ,    R    ,    S    ]
------------------------------------------------------------
```

3. Discussão dos Resultados

Ao utilizarmos um p_c menor que 100% nós modelamos um ambiente onde a execução do agente pode falhar gerando menores recompensas. Isso fica demonstrado nos valores menores atribuídos a cada estado para as iterações de valor e de política que consideraram p_c = 0.8 e $\gamma = 0,98$. Ademais, ao utilizarmos um gamma menor que 100% nós simulamos um ambiente em que o agente dará preferência às recompensas mais imediatas (dando menos prioridade a recompensas futuras) o que, no caso deste laboratório, acabou culminando em políticas com menos opções de ações. Cabe ressaltar que só faz sentido utilizar um gamma igual a 1 para casos em que haja um horizonte finito de possibilidades (problemas episódicos).