Laboratório 10 - Programação Dinâmica

Carlos R. A. Figueiredo[1]
Instituto Tecnológico de Aeronáutica, Laboratório de Inteligência Artificial para Robótica Móvel - CT-213. Professor Marcos Ricardo Omena de Albuquerque Máximo, São José dos Campos, São Paulo, 04 de junho de 2021.

[1]E-eletrônico: carlos.figueiredo@ga.ita.br

4.1 Implementação de Avaliação de Política

Para implementação da avaliação de política iterativa temos a seguinte equação:

$$v_{k+1}(s) = \sum_{a \epsilon A} \pi(a|s) r(s,a) + \sum_{a \epsilon A} \pi(a|s) r(s,a) p(s'|s,a) v_k(s')$$

Com isso é possível observar que teremos uma convergência para $v_{\pi}(s)$.

4.2 Implementação de Iteração de Valor

Para implementação de iteração de valor pode-se iterar diretamente sobre a equação de Bellman:

$$v_{k+1}(s) = max\left(r(s,a) + \sum_{s' \epsilon S} p(s'|s,a) v_k(s')\right)$$

Ela converge para $v_*(s)$.

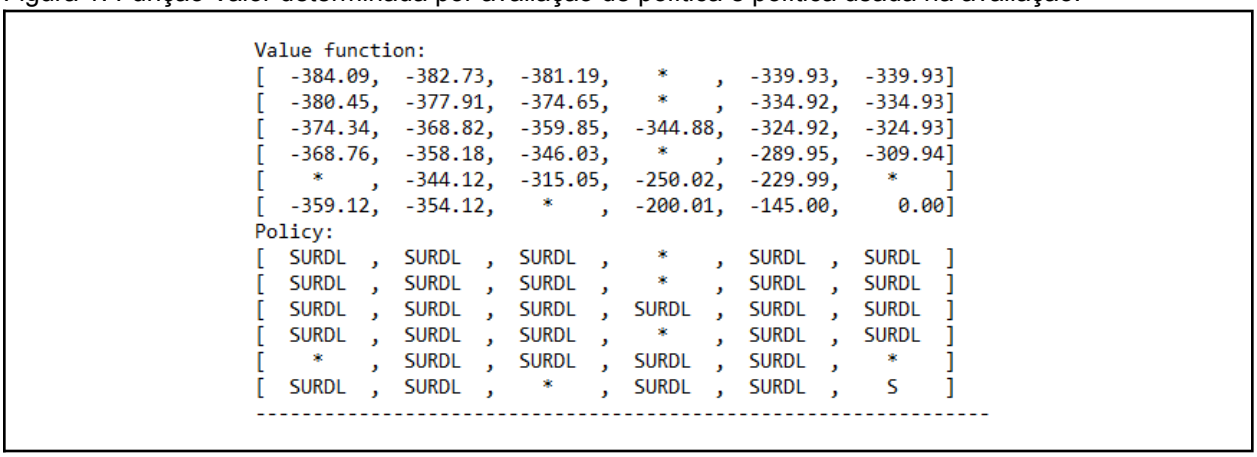
4.3 Implementação de Iteração de Política

Para essa implementação, basta fazer algo semelhante ao que foi feito no item 4.1, porém a cada 3 iterações(que é o `evaluations_per_policy` padrão) deve-se inserir uma política gulosa (greedy_policy).

4.4 Comparação entre Grid Worlds Diferentes

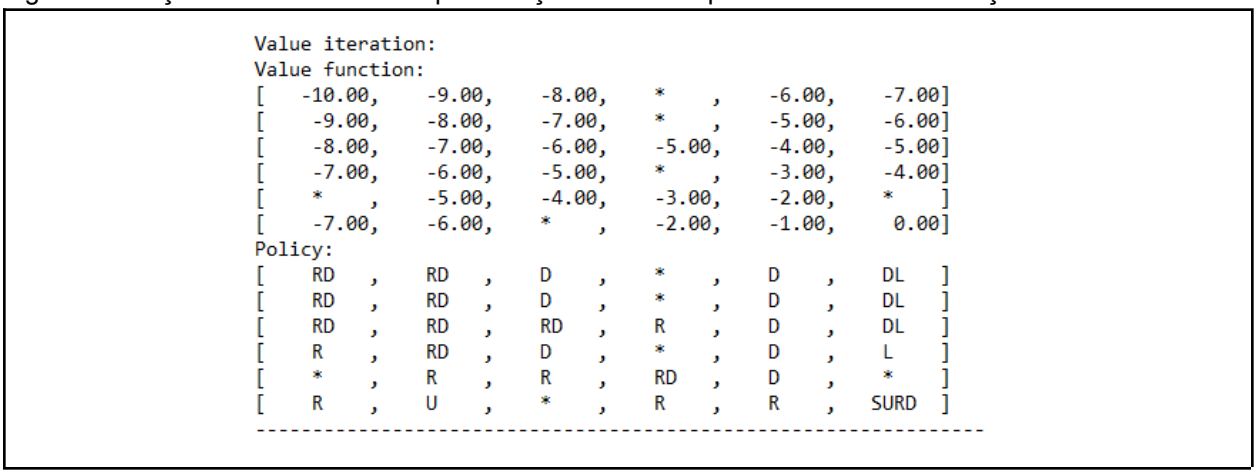
Para CORRECT_ACTION_PROB (p) = 1.0 e GAMMA = 1.0 encontramos o seguinte resultado:

Figura 1. Função Valor determinada por avaliação de política e política usada na avaliação.



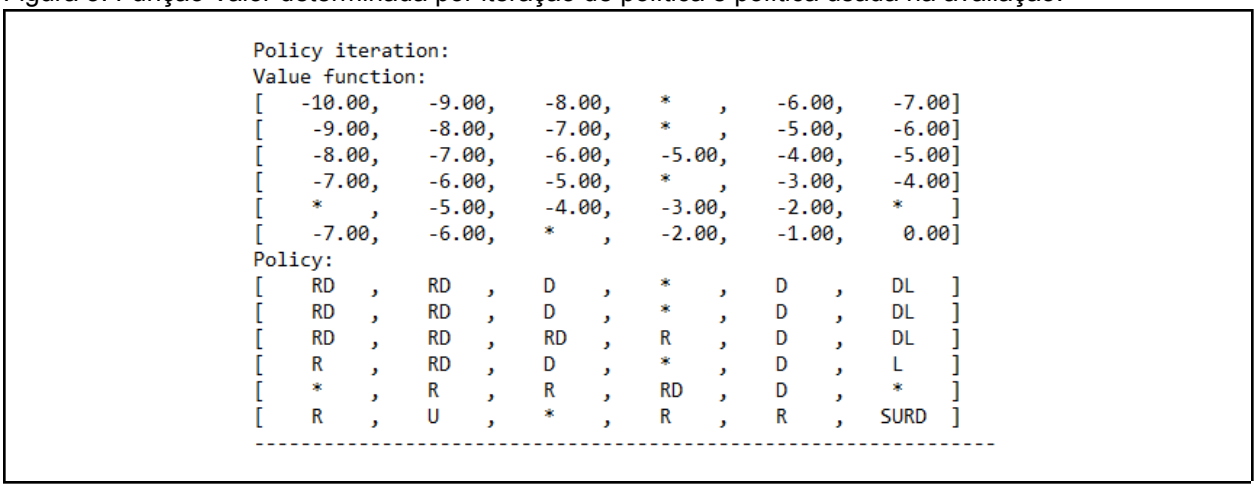
```
Value function:
[  -384.09,  -382.73,  -381.19,    *    ,  -339.93,  -339.93]
[  -380.45,  -377.91,  -374.65,    *    ,  -334.92,  -334.93]
[  -374.34,  -368.82,  -359.85,  -344.88,  -324.92,  -324.93]
[  -368.76,  -358.18,  -346.03,    *    ,  -289.95,  -309.94]
[    *    ,  -344.12,  -315.05,  -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. Função Valor determinada por iteração de valor e política usada na avaliação.



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

Figura 3. Função Valor determinada por iteração de política e política usada na avaliação.



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

Para CORRECT_ACTION_PROB (p) = 0.8 e GAMMA = 0.98 encontramos o seguinte resultado:

Figura 4. Função Valor determinada por avaliação de política e política usada na avaliação.

```
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. Função Valor determinada por iteração de valor e política usada na avaliação.

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

Figura 6. Função Valor determinada por iteração de política e política usada na avaliação.

```
Policy 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    ]
[    R    ,    D    ,    D    ,    R    ,    D    ,    D    ]
[    R    ,    D    ,    D    ,    *    ,    D    ,    L    ]
[    *    ,    R    ,    R    ,    D    ,    D    ,    *    ]
[    R    ,    U    ,    *    ,    R    ,    R    ,    S    ]
-----------------------------------------------------------------
```

É possível notar que os valores das figuras 2 e 3 são iguais, o que vai de acordo com a teoria, já que eles tendem a convergir para um valor ótimo. O mesmo ocorre com as figuras 5 e 6.

Com relação a comparação dos Grid Worlds percebe-se que a diminuição do CORRECT_ACTION_PROB de 1 para 0.8 insere uma probabilidade de ser o caminho ser escolhido de uma forma errada.

Já a diminuição do fator de desconto GAMMA de 1 para 0.98 adiciona mais imediatismo, ou seja, uma recompensa imediata passa a ter um valor maior dentro do cálculo das recompensas total para o futuro.

Por fim, conclui-se que o método utilizado garante a convergência, porém ele escala mal, ou seja, problemas mais complexos podem ser inviáveis de serem resolvidos por conta da complexidade e do tempo (tabela explode). Apesar dessa falta de escalabilidade, hoje em dia ainda existem alguns problemas que utilizam esse método para resolução dos problemas em conjunto com técnicas de redes neurais para aproximar o v(s).