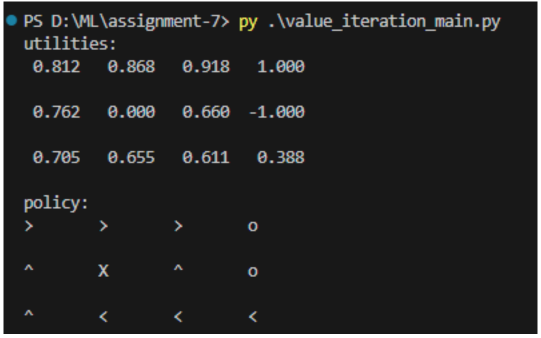
ML-7

siddharth bhagvagar

November 2024

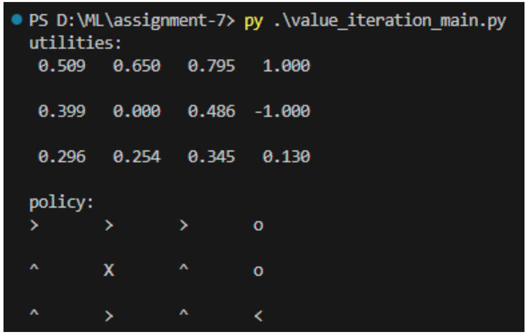
1 Task-1

When non-terminal-reward = -0.04 , γ = 1 and iteration = 20



```
PS D:\ML\assignment-7> py .\value_iteration_main.py
utilities:
  0.812   0.868   0.918   1.000

  0.762   0.000   0.660  -1.000

  0.705   0.655   0.611   0.388

policy:
>       >       >       o

^       X       ^       o

^       <       <       <
```

When non-terminal-reward = -0.04 , γ = 0.9 and iteration = 20



```
PS D:\ML\assignment-7> py .\value_iteration_main.py
utilities:
  0.509   0.650   0.795   1.000

  0.399   0.000   0.486  -1.000

  0.296   0.254   0.345   0.130

policy:
>       >       >       o

^       X       ^       o

^       >       ^       <
```

2 Task-2

2.1 Part-a

having a negative value for non-terminal states will encourage it to move the pieces more than having a positive or zero value. which can help it to learn the chess board properly and fully.

2.2 Part-b

the chess game is a long strategic game and generally lasts longer. Having a discount factor(gamma) close to 1 i.e. 0.9 ¿ x ¿ 1. can make big strategic play meaningful.

3 Task-3

3.1 Part-a

U((2,2),"UP") = 0.8 * 1 + 0.1 * -0.04 + 0.1 * -0.04 = 0.792
U((2,2),"DOWN") = 0.8 * -1 + 0.1 * -0.04 + 0.1 * -0.04 = -0.808
U((2,2),"LEFT") = 0.8 * -0.04 + 0.1 * 1 + 0.1 * -1 = -0.032
U((2,2),"RIGHT") = 0.8 * -0.04 + 0.1 * 1 + 0.1 * -1 = -0.032
U(2,2) = R(2,2) +γ [max(0.792,-0.808,-0.032,-0.032)]
U(2,2) = -0.04 + 0.9 * 0.792
U(2,2) = -0.04 + 0.9 * 0.792
U(2,2) = 0.6728

3.2 Part-b

U((2,2),"UP") = 0.8 * 1 + 0.1 * r + 0.1 * r
U((2,2),"UP") = 0.8 * 1 + (0.1 + 0.1) * r
U((2,2),"UP") = 0.8 * 1 + 0.2 * r
U((2,2), "LEFT") = 0.8 * r + 0.1 * 1 + 0.1 * -1
U((2,2), "LEFT") = 0.8 * r
if the UP is not optimal then it implies that the value of UP is less than any one of the other actions
For example,
$UP < LEFT$
$0.8 * 1 + 0.2 * r < 0.8 * r$
$0.8 + 0.2r < 0.8r$
$0.8 < 0.8r - 0.2r$
$0.8 < 0.6r$
$\frac{0.8}{0.6} < r$
$r > \frac{8}{6}$
$r > \frac{4}{3}$
$r > 1.33$