Long Nguyen

1001705873

1) value_iteration('environment2.txt', -0.04, 1, 20)

utilities:

0.812 0.868 0.918 1.000

0.762 0.000 0.660 -1.000

0.705 0.655 0.611 0.387

policy:

> > > o

^ x ^ o

^ < < <

value_iteration('environment2.txt', -0.04, 0.9, 20)

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) The reward for non-terminal states would be 0, meaning that the agent will make different moves without feedback on whether each move is good or bad. This is because having a reward for non-terminal states greater or less than 0 could introduce bias towards a certain playstyle, when chess is a game that values adaptability more.

The value of gamma would be 0.99 or any value close to 1. Having a value of gamma that is <1 makes the algorithm favor more immediate rewards; however, this would not be ideal in chess since chess is a game of setting up win conditions and trying to one up on your opponent. But having a value of gamma = 1, would make the algorithm focus too much on future rewards, which would be unfavorable in a game of chess since you must always stay on top of your opponent.

3a)

p((2,2) | (2,2), “up”) = 0.8

p((1,2) | (2,2), “up”) = 0.1

p((3,2) | (2,2), “up”) = 0.1

∑[p(s'|s,a)U[s']] = 0.8U(2,2) + 0.1(-1) + 0.1(1) = 0.8U(2,2)

p((2,2) | (2,2), “down”) = 0.8

p((1,2) | (2,2), “down”) = 0.1

p((3,2) | (2,2), “down”) = 0.1

∑[p(s'|s,a)U[s']] = 0.8U(2,2) + 0.1(-1) + 0.1(1) = 0.8U(2,2)

p((2,2) | (2,2), “left”) = 0.1

p((1,2) | (2,2), “left”) = 0.8

p((2,1) | (2,2), “left”) = 0.1, blocked stays in (2,2)

∑[p(s'|s,a)U[s']] = 0.8(-1) + 0.1U(2,2) + 0.1U(2,2) = -0.8 + 0.2U(2,2)

p((2,2) | (2,2), “right”) = 0.1

p((2,1) | (2,2), “right”) = 0.1, blocked stays in (2,2)

p((3,2) | (2,2), “right”) = 0.8

∑[p(s'|s,a)U[s']] = 0.8(1) + 0.1U(2,2) + 0.1U(2,2) = 0.8 + 0.2U(2,2)

U(2,2) = -0.04 + 0.9 * max{∑[p(s'|s,a)U[s']]}

For 0.8U(2,2):

U(2,2) = -0.04 + 0.9(0.8U(2,2)) = -0.143

For -0.8 + 0.2U(2,2):

U(2,2) = -0.04 + 0.9(-0.8 + 0.2U(2,2)) = -0.927

For 0.8 + 0.2U(2,2):

U(2,2) = -0.04 + 0.9(0.8 + 0.2U(2,2)) = 0.829

Since “right” has the highest utility, U(2,2) = 0.829

3b) Using the values found in 3a, let x = U(2,2)

“up” < “left”:

0.8x < -0.8 + 0.2x => 0.6x < -0.8 => x < -4/3

“up” < “right”:

0.8x < 0.8 + 0.2x => 0.6 < 0.8 => x < 4/3

* Cannot use “down” since it is similar to “up”

Using the Bellman Formula, for x <= -4/3

x = r + 0.9(-0.8 + 0.2x)

r = 0.82x + 0.72 = -3.64

For -4/3 < x <= 4/3:

$x = r + 0.9(0.8x)$

$r = 0.28x = 0.373$

For $x > 4/3$:

$x = r + 0.9(0.8 + 0.2x)$

$r = 0.82x – 0.72 = 3.64$

Therefore the range of r for which “up” is not optimal is $-3.64 <= r <= 3.64$