Machine Learning Reinforcement Learning Exploration

Dr. Alexjandro Daviano

Today I wanted to choose a reinforcement learning method and use it to implement a program. We will be able to do this based on the packages such tic-tac-toe and CRAN to help with the assignment. The package “tictactoe” will be useful to help the agent learn more about how to better improve the AIs in playing the game, through using the q learning and stimulate functions.

Methods and Results/interpretations

First, we will upload the CRAN and the tic-tac-toe packages to use the tic-tac-toe function to play tic tac toe with the AI. Once we get a feel for tic tac toe with the AI we will stop playing and add another AI into the game, so they can play together.

```
> ttt(ttt_ai(), ttt_ai())
    A B C
   ------
1| . . .
2| . . .
3| . . .

Player 1 (ttt AI) to play
action = C2
   A B C
  ------
1| . . .
2| . . X
3| . . .

Player 2 (ttt AI) to play
action = A2
   A B C
  ------
1| . . .
2| O . X
3| . . .

Player 1 (ttt AI) to play
action = A3
   A B C
  ------
1| . . .
2| O . X
3| X . .

Player 2 (ttt AI) to play
action = A1
   A B C
  ------
1| O . .
2| O . X
3| X . .

Player 1 (ttt AI) to play
action = B1
   A B C
  ------
1| O X .
2| O . X
3| X . .

Player 2 (ttt AI) to play
action = B2
   A B C
  ------
1| O X .
2| O O X
3| X . .
```

```
 Player 1 (ttt AI) to play
 action = B3
    A B C
   ------
 1| O X .
 2| O O X
 3| X X .

 Player 2 (ttt AI) to play
 action = C1
    A B C
   ------
 1| O X O
 2| O O X
 3| X X .

 Player 1 (ttt AI) to play
 action = C3
game over
    A B C
   ------
 1| O X O
 2| O O X
 3| X X X

 won by Player 1 (ttt AI)!
```

We then see how the gam goes with two undefined/random AIs. What we can do to improve the outcome of the results and get better reward over time Is to raise the level of the AI strength. The highest level in R is level 5 on a scale from one to five. We then play the two AIs against each other when both are at the highest strength level for R. We will want to see if there are any gains in the intelligence or complexity of the game to get better AI players.

```
> ttt(ttt_ai(level = 5), ttt_ai(level = 5))
   A B C
  ------
1| . . .
2| . . .
3| . . .

Player 1 (ttt AI) to play
action = A2
   A B C
  ------
1| . . .
2| X . .
3| . . .

Player 2 (ttt AI) to play
action = C2
   A B C
  ------
1| . . .
2| X . O
3| . . .

Player 1 (ttt AI) to play
action = A1
   A B C
  ------
1| X . .
2| X . O
3| . . .

Player 2 (ttt AI) to play
action = B1
   A B C
  ------
1| X O X
2| X . O
3| O . .

Player 1 (ttt AI) to play
action = B2
   A B C
  ------
1| X O X
2| X X O
3| O . .

Player 2 (ttt AI) to play
action = C3
   A B C
  ------
1| X O X
2| X X O
3| O . O

Player 1 (ttt AI) to play
action = B3
game over
   A B C
  ------
1| X O X
2| X X O
3| O X O

draw!
```

Now that we can see that the two AIs are strong enough to play against each other and draw, we want to see if we can strengthen them by customizing them as undefined AIs. We then use the simulate function to get better AIs that have learned from their environments by playing a lot of games. We then simulate the over 100 games for the AIs the first simulation, and then we will increase that to 250 games to train the AIs and see what the table will say of the results.

```
> res <- ttt_simulate(ttt_ai(), ttt_ai(), N = 100, verbose = FALSE)
> prop.table(table(res))
res
   0    1    2
0.16 0.59 0.25
> res <- ttt_simulate(ttt_ai(), ttt_ai(), N = 250, verbose = FALSE)
> prop.table(table(res))
res
    0     1     2
0.112 0.536 0.352
```

We then see above that the more we have the AIs play the more they learn. It seems that AI one is winning more to start but as we increase the game in the second simulations we see that the second AI is starting to catch up with its strength by increasing in the total reward or result by 10%. However, we see that we can probably gain better win results from the AIs if we train them a bit more. We know that simulating is one way to increase the ability of the AI, but we can also use the function q learn to train the AI with larger amounts of games/episodes at a time. With the q learn training this will help us to gain the advantage over a random AI in games by training one of the AIs to be stronger than the random AI.

```
res
   0    1    2
0.17 0.17 0.66

res2
   0    1    2
0.19 0.05 0.76
```

With the q learn training we can see that the AI we trained does significantly better than the random AI. We see on the first table the player 2 was stronger than the random AI by 66%, while we then see with 500 more simulations of training we gained 10% in strength.

Overall, we see that the simulations and training of our dataset from the environment of the game helps the AI to learn. The Environment was the tic-tac-toe game, while the objective here was to win as many games as possible with the AI. We then wanted to train our AI to be the strongest player in the game once we played a bit. The policy we were playing by

was the draw, win or loss of the tic-tac-toe game. The reward function was the winning of the game within 3 to 4 moves.

Our value function increased with the amount of time spent playing as well as the training and showed us that 76% of the time we could expect to win. After the training once we got to 76% this gave us a new policy of wining at least 65% of the time and up to 75% of the time. The policy was also to map the state to the actions, which were to input an X or an O to arrive at a winning outcome. The exploitation was when we would keep placing an X or an O in a learned winning place to win a game. While the exploration was finding the patterns of the random AI and learning them to then find the best new way to win the game.

Conclusion

Today we looked at reinforcement learning with different ways to test it. We used the tic-tac-toe game to look at the different ways we could build a winning AI. This exploration was very useful in being able to use training techniques to gain better rewards over time with an AI.

References

Arel, I. (2015). Reinforcement Learning in Artificial Intelligence. Retrieved from: http://web.eecs.utk.edu/~itamar/courses/ECE-517/notes/lecture1.pdf

Sutton, R., & Barto, A. (2015). Reinforcement Learning: An Introduction. 2 ed.