Reinforcement Learning and Dynamic Programing in Gomoku

Author:Yuhui Sun

PART 1:Analysis Of The Algorithm

This essay is mainly discussing how to implement the methods of artificial intelligence in the classical game Gomoku.The rule of the game is to have two players choosing one of them to play first and the other then.The first player uses the black points to place on the board,and the second one uses the white.The winner is the player who first places five points with the corresponding color in diagonals,horizons or verticals.Therefore,it is apparent that we can use reinforcement learning to create an AI bot to fight against the human.The AI bot that we need to create can be considered as the agent,and the opponent should be seen as the environment.Assuming that we play first,each time we place a point on the board,the opponent(the environment) will receive the signal and do an action(the reward).In the classical problems of it,an ordinary way is to use Q-learning as well as the neural networks to solve it.The formula of Q-learning is

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left(r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right)$$

In this formula we can see that st is the current state,at is the action chosen for the current state. Q(S,A) is a function to approximate the value for the current state with the action chosen.Then,$\alpha$is the parameter to determine the speed to adapt the Q(S,A) function. Using the formula,we can update the Q-value for current state and action depending on the Q-value for the next state.Additionally,we might use neural networks to evaluate Q(S,A),but in this passage we will use dynamic programming to implement this. We can first let Q(S,A) be the probability of the state S with action A to win this game for the black points,P1(S,A) be the probability to make action A with state S for black points,P2(S,A) be the probability to make action B with state S for white points.Q(S) is the probability to win the game with state S,and S' is the state after taking the action A,S'' is that state after taking the action B.Then we have this formula

$$Q(S, A) = \sum_B (P2(S', B) \cdot Q(S'')) \quad (1)$$

According to Bayesian Theory

$$Q(S'') = \sum_{\alpha} (P1(S'', a) \cdot Q(S'', \alpha)) \quad (2)$$

$$Q(S, A) = \sum_{B} \left(P2(S', B) \cdot \sum_{\alpha} (P1(S'', a) \cdot Q(S'', \alpha)) \right) \quad (3)$$

$$\sum_{\alpha} P1(S'', \alpha) = 1 \quad (4)$$

So we can use this formula to update Q(S,A).More specifically,we need to determine the boundary of Q(S,A).

$$Q(S, A) = 1 \, (if \;\; the \, agent \, wins \, after \, taking \, action \, A \, with \, state \, S) \quad (5)$$

$$Q(S, A) = 0 \, (if \;\; the \, agent \, loses \, after \, taking \, action \, A \, with \, state \, S) \quad (6)$$

However,what about P1(S,A) and P2(S,A) functions?How can we initialize them and update them?We can first sketch these problems.We cannot immediately solve this problem,but we can assume all the actions in each state have the equal possibilities.There is a significant theorem

$$\sum_{A} P1, 0(S, A) = 1, \sum_{A} P2, 0(S, A) = 1 \quad (7)$$

Therefore,

$$P1, 0(S, A) = \frac{1}{|A|}, P2, 0(S, A) = \frac{1}{|A|} \, (|A| \;\; is \, the \, number \, of \, actions) \quad (8)$$

And this is for round 0.

After using the formula (3),(8) to update Q(S,A),we need to think about the next step to process.Q(S,A) is the probability to win the game with state S and action A.So an intuitive way to update P1(S,A) and P2(S,A) functions is

$$P1_{n+1}(S, \; A) = \frac{Q_n(S, \; A)}{\sum_A Q_n(S, A)} \quad (9)$$

It is understandable that if one action A with S is more possible to win the game,then the agent will have more possibility to choose action A.Then,there is one more complex problem,how to update P2(S,A)?First,we need to specify the internal form of the state S.State S is a matrix to save what color of points should be in each entry.Assuming that the entry is 1 if it is black,2 if it is white,0 if it is empty.Then we define ~S to be the transformation of S,the rule is below

$$(-S)_{i,j} = 2 \ (S_{i,j} = 1), (-S)_{i,j} = 1 \ (S_{i,j} = 2), (-S)_{i,j} = 0 \ (S_{i,j} = 0) \quad (10)$$

So the formula should be

$$P2_{n+1}(S,A) = P1_{n+1}(-S,A) = \frac{Q_n(-S,A)}{\sum_A Q_n(-S,A)} \quad (11)$$

So after implementing the formula for n rounds,we can get the expected parameter of P1(S,A) and P2(S,A).Overall,we can summarize the strategy to deal with this problem,the data that we need to calculate and collect the Q(S,A) for Nth-round,we first assume the P1(S,A) and P2(S,A) to be equal in each action,then in the following round,we try to let Q(S,A) be more accurate.The last step is to export the data and use it to predict the policy we when the AI bot(agent) interact with the opponent(environment).For each state S,we have several actions to choose.Depending on Q(S,A) data we collect,we can choose the action A that has the biggest Q(S,A).

PART 2:Testing The Accuracy Of The Algorithm

First,we can simplify this problem to reduce the time complexity to solve this problem.In the classical problems,we need to find five continuous points with the corresponding color,and we replace 5 by 3.

If there is a 3*3 grid,we can run code to calculate the data using the strategy above.The code is listed this URL link: [yuhui15/reinforcement-learning (github.com)]

The number of rounds to be set is 3.We can pick up some data from the whole data to see the tendency.

$$s = \begin{pmatrix} 1 & 1 & 2 \\ 2 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

This is the state that we choose to find out the tendency.

[1,1]	[1,2]	[1,3]	
-1	-1	-1	round 1
-1	-1	-1	round 2
-1	-1	-1	round 3

[2,1]	[2,2]	[2,3]	
-1	-1	0.666667	round 1
-1	-1	0	round 2
-1	-1	0	round 3

[3,1]	[3,2]	[3,3]	
0.666667	1	1	round 1
0.194444	1	1	round 2
0.118056	1	1	round 3

When find out the best solution to place the point without coding,it is trivial to see that we can place the black point in [3,2] or [3,3] to win the game.Depending on the table,we can apparently see that when the round increases,Q(S,[3,2]) increases and Q(S,[3,3]) increases,and all Q-value with the rest of the actions decreases,it matches the intuition and help us to find out the best solution.