Inductive Programming: Tutorial 8
Game Strategy Induction

Stephen Muggleton

The aim of this tutorial is to help you understand concepts in Lecture 8, involving Game Strategy Induction.

Question 1

1. What is a recent approach for machine learning strategies for complex board games?
2. What has been achieved by this approach?
3. What are two limitations of this approach?
4. Explain why it is valuable to make benchtest comparisons on Minimax evaluable games.

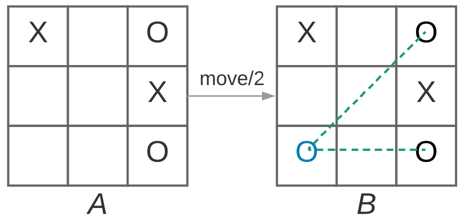
Solution

1. A recent approach is Deep Reinforcement Learning.
2. World-class play for Go, Chess, Checkers. This has been achieved by Deep Mind's AlphaGo (2016) and AlphaZero (2018).
3. Poor Data Efficiency and lack of Human Comprehensibility.
4. Minimax evaluable games are those in which there are sufficiently few board states that a minimax look-up table can be efficiently constructed. In this case minimax regret can be used as an absolute measure for performance comparisons of different learning systems.

Question 2

1. Illustrate a 2-ply minimax optimal move in Noughts-and-Crosses.
2. Show some general Prolog rules which explain how to make such a move.

Solution

1. 

2.
```
win_2(A,B):-win_2_1_1(A,B),not(win_2_1_1(B,C)).
  win_2_1_1(A,B):-move(A,B),not(win_1(B,C)).
        win_1(A,B):- move(A,B),won(B).
```

Question 3

1. What is the Credit Assignment problem?

2. How does Reinforcement Learning deal with the credit assignment problem?

3. How many board states are there in Noughts-and-Crosses, Chess and Go?

4. What does this imply about PAC learnability for these games?

Solution

1. The credit assignment problem concerns how to attribute the game outcome for a playing strategy to the individual moves carried out.

2. Assign rewards to individual moves based on a delay function. Rewards used to update parameters across all board states in a game.

3. The number of board states for Noughts-and-Crosses is 10^5; Chess is 10^{45}; Go is 10^{100}.

4. Reinforcement learning will be ineffective for games such as Chess and Go without a compact description of the learned strategy. Such a description cannot involve a parameter for each board state.

Question 4

1. How does MIGO solve the Credit Assignment problem.

2. State the key Lemma which supports this solution.

Solution

1. Based on using two theorems, MIGO extracts positive examples for win_i and $draw_i$ moves from games in which it wins or draws against a minimax opponent in Ply i.

2. When any player P_1 plays against a minimax opponent P_2 the outcome of P_1 monotonically decreases during a game.

Question 5

1. What are a) minimax reget and b) cumulative minimax regret?

2. What is required to efficiently calculate minimax regret?

Solution

1. a) The minimax regret of game G is the difference between the minimax outcome of the initial position in G and the actual outcome of G.

2. It is necessary to construct a database of minimax values for all board game states.

Question 6

1. What was the null hypothesis for the first experiment?

2. What was the outcome of the experiment?

3. Why do you think this outcome happened?

4. Name two weaknesses of the MIGO approach relative to reinforcement learning.

5. Name three strengths of the MIGO approach relative to reinforcement learning.

Solution

1. MIGO cannot converge faster than MENACE/HER, Q-learning and Deep Q-learning for learning optimal two-player game strategies.

2. The null hypothesis was refuted.

3. MIGO was able to find a compact, near-optimal logic program for playing Noughts-and-Crosses and Hexapawn.

4. a) MIGO's compact strategies are inefficient when scaled to larger board games, b) MIGO assumes a minimax opponent.

5. a) Improved Data Efficiency since MIGO converges to near-optimal play faster, b) MIGO produces readable strategies with potential to teach humans to play better, c) MIGO supports transfer learning among two person games.