人工智慧概論 HW3 報告

110550088 李杰穎

April 29, 2022

1 Adversarial Search

1.1 Implementation of Minimax and Expectimax Algorithms

Code 1: Minimax Algorithm

```
class MinimaxAgent(MultiAgentSearchAgent):
    """
    Your minimax agent (par1-1)
    """
    def getAction(self, gameState):
        """
        Returns the minimax action from the current gameState using self.depth
        and self.evaluationFunction.

        Here are some method calls that might be useful when implementing minimax.

        gameState.getLegalActions(agentIndex):
        Returns a list of legal actions for an agent
        agentIndex=0 means Pacman, ghosts are >= 1

        gameState.getNextState(agentIndex, action):
        Returns the child game state after an agent takes an action

```

```
        gameState.getNumAgents():
        Returns the total number of agents in the game

        gameState.isWin():
        Returns whether or not the game state is a winning state

        gameState.isLose():
        Returns whether or not the game state is a losing state
        """
        "*** YOUR CODE HERE ***"
        # Begin your code
        actions = gameState.getLegalActions(0)
        candidates = []
        for action in actions:
            candidates.append((action, self.minimax(gameState.getNextState(0, action),
            ↪  self.depth-1, 1, False)))
        action, _ = max(candidates, key=lambda item: item[1][1])
        # print(f"action: {action}")
        return action
        # End your code
    def minimax(self, gameState, depth, agentIdx, maximize):
        if gameState.isWin() or gameState.isLose() or (depth == 0 and agentIdx == 0):
            return (gameState, self.evaluationFunction(gameState))
        actions = gameState.getLegalActions(agentIdx)
        candidates = []
        if maximize:
            for action in actions:
                candidates.append(self.minimax(gameState.getNextState(agentIdx,
                ↪  action), depth-1, 1, False))
            stateScore = max(candidates, key=lambda item: item[1])

        elif agentIdx < gameState.getNumAgents()-1:
            for action in actions:
                candidates.append(self.minimax(gameState.getNextState(agentIdx,
                ↪  action), depth, agentIdx+1, False))
            stateScore = min(candidates, key=lambda item: item[1])
        else:
            for action in actions:
                candidates.append(self.minimax(gameState.getNextState(agentIdx,
                ↪  action), depth, 0, True))
            stateScore = min(candidates, key=lambda item: item[1])
```

```
56
57          return stateScore
```

1.2 Minimax 與 Expectimax 的比較

2 Q-learning

3 DQN