Q-LEARNING PLAYER

Implementation , results & future works

Q-Learning Algorithm

- Q value update function

$$Q(s_i, a_i) \leftarrow (1-\alpha) \times Q(s_i, a_i) + \alpha \left[r_i + \lambda \max_{a_{i+1}} Q(s_{i+1}, a_{i+1}) \right]$$

Where

- $s_i / a_i / r_i$ is the current state/action/reward,
- s_{i+1} / a_{i+1} is the next state/action;
- α is the learn rate and λ is the discount factor to the long term payoff.

- Action decision making function based on Boltzmann distribution

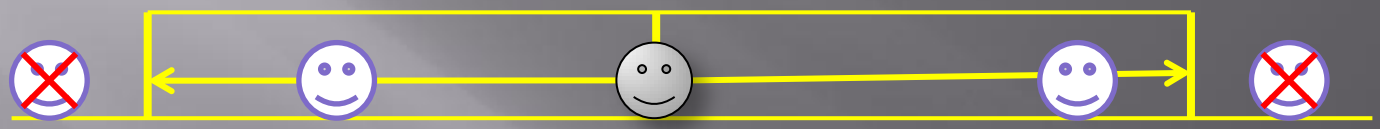
$$p(a_i \mid s) = \frac{e^{Q(s,a)/T}}{\sum_{a_i \in A} e^{Q(s,a_i)/T}}$$

Where

- A is action set – specifically cards in hand for current turn
- T is the temperature parameter which adjust the exploration of learning.

Learn to Play Cards

- State representation
  - Only considered the relative position with closest opponents in both forward and backward direction

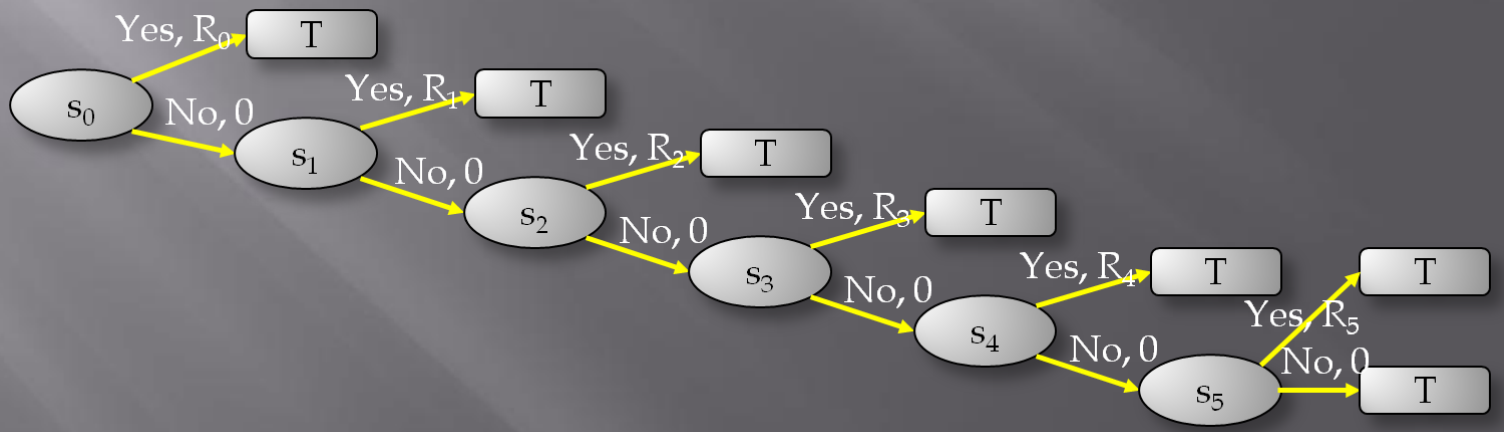


  - The range of distance between the player and opponents was set to ±10, so, the total size of state space is 12*12 = 144
  - The position of Derelicts were not be considered
  - Combine the relative position states with turn, then the size of state space was extended to 6*144 = 864
  - Then, the game can be treated as many MDPs with 6 levels.

- Action representation
  - When learn to update Q values, 26 possible cards can be played, so, 26 possible actions;
  - When choose card to play, then the action set is cards in hand;
  - Since the game is imperfect information game, the cards held by opponents and which card the opponents will play are totally unknown. Also, it is hard to use fictitious play to predict opponents' strategies. So, I just make the RL player ignore the actions taken by the opponents by treating them as un-stationary environment.

Learn to Use ES card

- State representation
  - States are exactly represented by the tuple {turn, closestShipDirection, playedCardType, playedCardValue}
  - The size of state space is 6x2x2x10 = 240.
- Action representation
  - Only 2 actions available, Use/not Use
  - The trick of learning to use ES card is how to balance immediate reward and long term reward since the player can only use ES card one time in a round.



Performance of RL player

Run the game 30000 times with different gameAI configurations

Type of gameAI	No. of Win
randAI	4022
RLAI	7732
symbolicAI	8986
symbolicAI	9260

Type of gameAI	No. of Win
randAI	4416
randAI	4399
RLAI	9497
symbolicAI	11706

Type of gameAI	No. of Win
randAI	5769
randAI	5825
randAI	5783
RLAI	12623

RL player can significant outperform random players but still be weaker than symbolic players.

Possible Future Works for RL player

- For learning to play cards
  - Better state representation, current state representation method is still weak to full represent the real states for play. Some neural network based state representation methods can be introduced.
  - The position of Derelicts should be considered
  - Possibility of using joint-action learning since the reward of each player is significantly effected by the actions taken by other players.