

進捗報告

1 やったこと

- GPT 4 同士の対戦
- 簡易的な Emperor Card 環境の適用

2 GPT4 同士の対戦

元ある実装は, LLM agent vs 深層強化学習による学習済みモデルの対戦のみの実装であったため, LLM agent 同士で対戦できるように改変した. お互いに 50 枚ほどチップを持った状態からスタートし, Luduc Hold'em を複数ゲームプレイした. 15 ゲーム後の終了時には 1 方のチップが 73, もう 1 方のエージェントのチップが 30 となっていた. Agent は Belief, Pattern, Plan など様々な LLM モジュールで構成されているが, Agent の対戦相手の行動を分析する Pattern を例に挙げて 1 ゲーム目と, 15 ゲーム目の出力結果を比較する.

GPT 4 の Pattern の出力例 (1 ゲーム目スタート時) —

sorry, but I can't assist with that.

GPT 4 の Pattern の出力例 (15 ゲーム目スタート時)

"Player_LLM_1's game pattern:

When Player_LLM_1 holds a King:

- In the 1st round, if Player_LLM_1 is the first to act, he would like to fold (probability 100%).
- If Player_LLM_1 sees the opponent raise, he would like to fold (probability 100%).

When Player_LLM_1 holds a Queen:

- In the 1st round, if Player_LLM_1 is the first to act, he would like to raise (probability 100%).
- If Player_LLM_1 sees the opponent raise, he would like to call (probability 100%).
- In the 2nd round, if the public card is a King or Jack, Player_LLM_1 would like to raise (probability 100%). If the opponent raises, he would like to call (probability 100%).

When Player_LLM_1 holds a Jack:

- In the 1st round, if Player_LLM_1 is the first to act, he would like to call (probability 100%).
- If Player_LLM_1 sees the opponent raise, he would like to call (probability 100%).
- In the 2nd round, if the public card is a King or Queen, Player_LLM_1 would like to call if the opponent raises (probability 100%).

Player_LLM_1's guess on my game pattern:

- In the 1st round, when Player_2 holds a Queen or King, Player_LLM_1 believes Player_2 would like to raise (probability 100%).
- In the 2nd round, when Player_2 holds a Queen or King, and the public card is a Jack or Queen, Player_LLM_1 believes Player_2 would like to raise (probability 100%).

Strategy Improvement:

When I hold a King and the public card is a Queen or Jack:

- If I see Player_LLM_1 raise, I would like to call.

When I hold a Queen and the public card is a King or Jack:

- If I see Player_LLM_1 raise, I would like to call.

When I hold a Jack and the public card is a King or Queen:

- If I see Player_LLM_1 raise, I would like to call.

This strategy exploits Player_LLM_1's tendency to raise when he holds a Queen or Jack, and his tendency to fold when he holds a King. By calling his raises, I can increase the pot and potentially win more chips."

文中に出てくる Player_LLM_1 は対戦相手の名前である。ゲームを進めていくにつれて対戦相手のパターンを記憶できていることがわかる。また、1 ゲーム目には Pattern を判別できるような過去の対戦の記憶が存在しないため、Pattern はうまく生成できていないとわかる。

2.1 簡易的な Emperor Card の実装

環境へ適用させるためのインターフェースの作成、環境への適用で沼ってしまったため、一度簡単にした条件で実験を回している。

2.2 簡易的な Emperor Card 環境

ルールは基本的に原作に準拠して以下の通り実装した。掛け金の場合は原作のように身体の一部を賭けることはできないのでルールを定めた。

- プレイヤーは 2 人で奴隷サイド、皇帝サイドに分かれそれぞれ手札は市民 4 枚と奴隷 1 枚、皇帝 1 枚の計 5 枚の手札を持つ

- カードを出し合い, 勝負がついたら 1 ラウンド終了, 3 ラウンドごとに奴隷と皇帝サイドを入れ替わり計 12 ラウンド実行する.
- ゲーム開始時に 100 チップごと各プレイヤーに配る. プレイヤーは各ラウンド始めに 1 持っているすべてのチップの枚数分のチップを Bet する.
- 市民 vs 奴隷, 皇帝 vs 市民で勝利した場合は賭けた分払い戻しされ, 奴隷 vs 皇帝で勝利した場合は賭けた分の 5 倍払い戻しされる. 敗北した場合は賭けた分手持ちのチップから失われる.
- 12 ラウンド終了後残りチップが多いほうが勝利, または手持ちのチップが 0 になったプレイヤーの敗北

2.3 状態空間と行動空間

簡略化した部分として, Agent の行動として賭けのみを選択する形にした. Agent の状態空間は以下のよう

に定義した.

- プレイヤー, 相手プレイヤーの残りチップ数
- プレイヤー, 相手プレイヤーのサイド (皇帝 or 奴隷)
- 現在のラウンド数
- 実験開始からのゲーム数

Agent の行動空間は 1 (現在持っているチップ数) の離散的な数字列であり, Action = 1 と選択するとそのラウンドは 1 枚チップを賭ける動作をする.

2.4 game_config について

実験環境ではゲームのルールと状態空間の各変数が何を表しているかをあらかじめプロンプトとして指定しておく必要がある. 以下のように指定した.

Game Rule

Emperor Card is a card game for two players that appears in a famous Japanese manga. Players are divided into slave side and emperor side, and each player's hand consists of 5 cards: 4 citizen cards, 1 slave card if the player is on the slave side and 1 emperor card if the player is on the emperor side.

Action: The action is represented by a number, which indicates the number of chips that can be betted; choosing 1 is equivalent to betting 1 chip.

Setup: The side is randomly determined at the beginning of the game, and both players have 100 chips.

GamePlay: The game consists of 12 rounds, with the 'Emperor' and 'Slave' sides alternating between players every three rounds. In each round, bets are placed in order, starting with the player on the slave side. Bets may range from 1 to all chips in hand. After each player finishes betting, a shuffled hand is opened.

Winning Conditions: If the 'Emperor' faces a 'Citizen', the Emperor side wins. If the 'Slave' faces the 'Emperor', the Slave side wins with a higher payoff, typically five times the bet amount. If the 'Slave' faces a 'Citizen', the Slave side loses. if the 'Citizen' faces a 'Citizen', next card opens.

End Game: As for refunds, the losing player loses all his or her chips, the winning player gets the number of chips he or she played for if he or she wins with the combination of Citizen and Emperor or Citizen and Slave, and the winning player gets five times his or her chips for winning with the combination of Slave and Emperor. The game continues for 12 rounds or until one player runs out of chips; the winner is the player with the most chips at the end of 12 rounds or at the end of the game.

Observation Rule

The observation is a dictionary. The main observation space: 'raw_legal_actions' which holds the legal moves, described in the Legal Actions Mask section. 'game_num' is the idx of game now, 'player_money', 'opponent_money' indicate the remaining chips of the player and the opposing player. 'player_side', 'opponent_side' indicate whether the player and the opposing player are slave-side or emperor-side. 'current_round' indicates the current round in the game., please analysis what is your best combination now and potential combinations in future with probability (number) reasonably and accurately step by step.

3 課題

現在 GPT4 同士で対戦している. 4 ゲーム目の Agent の Action を決定する Plan モジュールの出力は以下のようになっていた.

E カード環境 4 ゲーム目 Plan

Plan 1: Bet a small amount (1 chip)

- If Player_LLM.1 holds a Citizen card (90% probability), he will likely bet a small amount (80% probability). If this happens, I will win the round and gain 1 chip. If he bets a moderate amount (20% probability), I will still win the round and gain 1 chip.
- If Player_LLM.1 holds a Slave card (10% probability), he will likely bet a small amount (90% probability). If this happens, I will lose the round and lose 1 chip. If he bets a moderate amount (10% probability), I will still lose the round and lose 1 chip.

Plan 2: Bet a moderate amount (50 chips)

- If Player_LLM.1 holds a Citizen card (90% probability), he will likely bet a small amount (80% probability). If this happens, I will win the round and gain 50 chips. If he bets a moderate amount (20% probability), I will still win the round and gain 50 chips.
- If Player_LLM.1 holds a Slave card (10% probability), he will likely bet a small amount (90% probability). If this happens, I will lose the round and lose 50 chips. If he bets a moderate amount (10% probability), I will still lose the round and lose 50 chips.

Plan 3: Bet a large amount (100 chips)

- If Player_LLM.1 holds a Citizen card (90% probability), he will likely bet a small amount (80% probability). If this happens, I will win the round and gain 100 chips. If he bets a moderate amount (20% probability), I will still win the round and gain 100 chips.
- If Player_LLM.1 holds a Slave card (10% probability), he will likely bet a small amount (90% probability). If this happens, I will lose the round and lose 100 chips. If he bets a moderate amount (10% probability), I will still lose the round and lose 100 chips.

The expected chips gain for each plan is:

- Plan 1: $(0.9 * 1 * 1) - (0.1 * 1 * 1) = 0.8$ chips
- Plan 2: $(0.9 * 50 * 1) - (0.1 * 50 * 1) = 40$ chips
- Plan 3: $(0.9 * 100 * 1) - (0.1 * 100 * 1) = 80$ chips

Ranking the plans by expected chips gain, we get:

1. Plan 3
2. Plan 2
3. Plan 1

Therefore, the best plan to execute is Plan 3: Bet a large amount (100 chips).

4 今後の課題

環境の拡張をする。

参考文献

表 1: 定義する予定の状態空間

状態説明	次元数	最小値	最大値
各プレイヤーの残りチップ数	2	0	プレイヤーの持っているチップの枚数
各プレイヤーのサイド	2	0	1
各プレイヤーが現ラウンドで賭けた枚数	2	1	プレイヤーの持っているチップの枚数
自分の手札 1	3	0	2
自分の手札 2	3	0	2
自分の手札 3	3	0	2
自分の手札 4	3	0	2
自分の手札 5	3	0	2

表 2: 定義する行動空間

行動説明	次元数
チップを賭ける	持っているチップの枚数
手札 1 ~ 5 を出す	5