

駒の動きの時系列データを用いて駒色を予測するガイスターAIの構築

Geister AI Predicting the Colors of Pieces Based on Time Series Data of Piece Movements

佐藤 光希¹ 穴田 一¹

Kouki Sato Hajime Anada

1. はじめに

近年、囲碁や将棋などの完全情報ゲームにおいて、AIは人間のトッププレイヤーを上回る実力を獲得している。一方、不完全情報ゲームではトッププレイヤー並みの実力を持つAIが実現されていない。これは対戦相手の所持している手が見えず、相手の状態や状況の予測が難しいことや、駆け引きが重要になることが原因であると考えられる。

不完全情報ゲームでトッププレイヤー並みの実力を持つためには、AI自身が駆け引きを行い、自分が有利になる人間のような行動をする必要がある。これを実現することで、実際に駆け引きが必要となる場面で状況に合わせた駆け引きを行うAIの研究に活かせるのではないかと考えた。

本研究で対象とする二人不完全情報ゲームのガイスターには、トッププレイヤーの棋譜が存在しないため棋譜を用いた学習は難しく、初心者相手にさえ勝てるほどの強いAIは存在しない。この原因の一つとして、人間の騙し行動をAIが見抜くことは難しいことが考えられる。

そこで本研究では、ガイスターを用いて、駆け引きを行う上で重要である相手の駒色とゲーム上での騙し行動である駒色を偽る動きをした駒について時系列データから推定するAIを構築する。

2. ガイスター

図1のようにガイスターの盤面は6×6で、盤面の四隅にはそれぞれ脱出マスが存在している。各プレイヤーはそれぞれ赤駒と青駒を4つずつ、計8つ所持している。ゲーム開始時、これらの駒を相手に色が分からないように盤面手前中央の2×4のマスに自由に配置する。図1に初期配置の例を示す。矢印は出口を表している。手番では自分の駒のどれか1つを自分の駒が存在しない上下左右どれかのマスに1マス動かさなければならない。この時、相手の駒が存在するマスに自分の駒を動かした場合、相手の駒を取り、その際にその駒の色を知る。

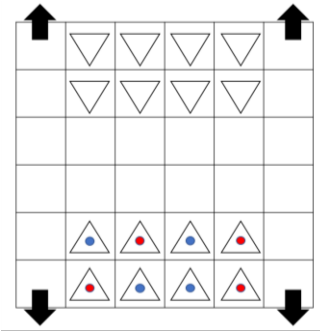


図1 ガイスターの盤面

各プレイヤーは3つの勝利条件のうち、どれか1つを満たすことでゲームに勝利する。

- 自分の青駒を脱出マスから脱出させる
- 相手の青駒を全て取る
- 自分の赤駒を全て取らせる

ガイスターでの駆け引きでは、勝利条件を満たすために相手に自分の駒色を間違えて認識させること、相手の駒色を正しく推定することが重要である。

3. 既存研究

ガイスターにはトッププレイヤーの棋譜が存在しないため、教師あり学習は不可能である。そこで木村らは強化学習の手法の一つであるAlphaZeroのアルゴリズムを用いて、相手の駒色分かる完全情報状態でガイスターAIを作成した。AlphaZeroのアルゴリズムを用いることで棋譜が無くても学習を行うことができる。しかし、実際のガイスターは相手の情報が分からない不完全情報状態であるため、実際の試合では用いることができない。そこで木村らはこのガイスターAIに、末續らが提案した相手の駒色の推定[1]を組み込む手法を提案した[2]。相手の駒色の推定方法は、全ての敵駒が青らしさを初期値0で保持し、手番ごとに駒の動きに応じて以下の表1にある点数を用いてその値を更新していく。これにより、推定した青らしさが正しいと仮定することによって、各エージェントが完全情報ガイスターとしてゲームを進めることができる。

¹ 京都市大学大学院 総合理工学研究科
Graduate School of Integrative Science and Engineering,
Tokyo City University Graduate School

表 1. 青らしさの更新値

駒のふるまい	更新値
前進し, 接敵数が増えた	-1.5
前に移動して, 接敵数が 0 になった	+4.0
...	...

4. 提案手法

まず相手の駒色が分かっている状態で AlphaZero を用いてガイスターAI を作成する. この際, 学習で行う自己対戦数は 500 回行った. 既存研究では, 推定内容を赤駒と青駒の 2 つとしていた. しかし, ガイスターでは自分の駒色を偽る動きをすることが戦略の一つであるため, 本研究では推定内容を赤駒, 青駒, 青駒のふりをした赤駒, 赤駒のふりをした青駒の 4 つとし, LSTM(Long Short-Term Memory) で推定を行う. これを AlphaZero で作成したガイスターAI に組み込み, 駒色が分かっているものとしてゲームを行う. LSTM を用いる理由は, 相手の駒の動きの時系列データを使うことで騙しの動きをしている駒の判別ができると考えたからである. LSTM では, 入力データに各マスの駒の有無と自分と相手の駒の取得状況の計 40 種類のデータを使う. また 5 ターン分のデータから次のターンで動いた駒色を推定する. 教師データは私たちが実際にガイスターをプレイした際の棋譜を用いる. これはガイスターの棋譜が存在しないことや, 大会で上位の実力を持つガイスターAI でも人間の初心者相手には勝てないことから初心者の棋譜も有効だと考えたからである.

LSTM での出力結果が青駒もしくは赤駒のふりをした青駒と出力した場合に駒を取得する行動をする.

5. 結果

表 2 に駒色が分かっている状態で, AlphaZero で作成したエージェントが行動した際の試合結果, 表 3 に駒色推定を加えたエージェントの対戦結果を示す. RAND はランダムエージェント, MCTS はモンテカルロ木探索エージェントを表している. MCTS は相手の駒色が変わらない状態でプレイアウトを行い, 行動を選択している. 勝率は勝利数 ÷ (勝利数 + 敗北数) で計算したものである. また勝敗のつき方についての結果を表 4, 表 5 で示す.

表 2, 表 3 から, 駒色推定を加えた場合の方が MCTS 相手への勝率が高いことが確認できる.

表 4 から RAND 相手への勝ち方は青駒を取る勝ち方が一番多く, 負け方は青駒脱出が一番多い.

表 2. 駒色が分かっているエージェントの対戦結果

	対 RAND	対 MCTS
勝利	93	27
敗北	0	40
引き分け	7	33
勝率	100%	40.3%

表 3. 駒色推定を行うエージェントの対戦結果

	対 RAND	対 MCTS
勝利	91	42
敗北	8	47
引き分け	1	11
勝率	91.9%	47.2%

表 4. 駒推定エージェント対ランダム対戦結果詳細

勝ち方		負け方	
青駒を取る	42	青駒を取られる	3
赤駒を取られる	9	赤駒を取る	0
青駒脱出	40	青駒脱出	5

表 5. 駒推定エージェント対 MCTS の対戦結果詳細

勝ち方		負け方	
青駒を取る	12	青駒を取られる	46
赤駒を取られる	13	赤駒を取る	0
青駒脱出	17	青駒脱出	1

表 5 から MCTS 相手への勝ち方は青駒脱出が一番多く, 負け方は青駒を全て取られる負け方が一番多い.

6. おわりに

駒色の推定が上手くいっている場合, 青駒を取る勝ち方が多くなり, 赤駒を取られる負け方が少なくなるはずだが, 回数を確認すると駒色推定が上手くいっていると思う. これは時系列データから駒色を偽る行動をした駒について推定したことにより, 作成したガイスターAI が人間に近い考え方で行動を選択できたのではないかと考える.

発表では, 追加で作成したモデルでの赤駒と青駒のみの 2 値分類推定した場合の結果を出力し, 本研究で出力した 4 値分類での結果と比べ, 騙しの動きを導入した有効性について述べる.

参考文献

- [1] 末續鴻輝, 織田祐輔, 機械学習を用いないガイスターの行動アルゴリズム開発, GAT2018 論文集, pp. 13–16 (2018)
- [2] 木村勇太, 伊藤毅志, 深層強化学習を用いたガイスターAI の構築, 電気通信大学, ゲームプログラミングワークショップ論文集, vol. 2019, pp. 130–135 (2019)