

ボードゲームにおける対戦戦略の適応的切り替え選択に関する研究 -オセロを用いて-

2016SE068 五月女 勇毅

指導教員：野呂 昌満

1 はじめに

ゲーム作品において開発側があらかじめ設定した難易度の中から、プレイヤー側が適切と思うものを選択し遊ぶという形式が多い。しかしあらかじめ設定された難易度が全プレイヤーに対応しているとは限らない。また、プレイヤーにとって選択した難易度が適当であるかプレイヤー本人にしか判断できない。ゲーム側がプレイヤーの練度を測りまたそれを難易度設定に活かしている作品は存在する。しかし一定回数失敗をする、負けるなどの過程を踏んだ後、難易度変更を提案するというものである。

これらのことから、CPU モデルごとの強さが難易度設定に用いられることが多いと考えた。

既存のゲーム難易度設定における問題点は、以下の2点である。

1. 個々のプレイヤーに対応した難易度設定
2. プレイヤーの練度に合わせて再設定

ストレスなくゲームをするためには、プレイ中にリアルタイムでプレイヤーの練度に合わせた難易度の自動設定を実現することが、必要であると考えた。

本研究の目的は、リアルタイム性を有したまま、プレイヤーの練度に合わせて難易度変更を行うことのできるアーキテクチャの設計、提案である。

既存の機械学習を応用してプレイヤーの練度を測り、CPU モデルの動作を決定するためのアーキテクチャを提案することで、ゲーム開発における難易度設定の効率化を図れると考えた。

今回の研究には、次の二つの理由からオセロを題材とする

1. ボードゲームは知名度が高く、その中でもオセロはルールが単純であるため
2. 既存の機械学習を応用できるため

これらのことから、オセロを題材とし、既存技術を応用することで、プレイ中の練度測定と、難易度の自動変更の実現を目指す。

研究課題は、以下の3つである。

1. プレイヤーの練度、CPU モデルの最適手の定義
2. 最適手の学習方法の定義
3. 前提、要求を考慮したアーキテクチャの設計指針を定義し、アーキテクチャ設計の提案を行う

これらの研究課題を解決することで、オセロ対戦中に CPU モデルは随時、プレイヤーの練度に合わせた難易度を予測、選択することが出来る。

2 背景技術

2.1 ゲーム木

完全情報ゲームは、お互いがどの手を打ったかによってどのような局面が出現するかを場合分けしていくことでゲーム展開を樹形図にできる。このように現在の局面から出現するすべての局面の関係をゲーム木という。

2.2 $\alpha\beta$ 法

二人零和有限確定完全情報ゲームの探索アルゴリズムとして最適とされている方法である。静的評価関数を用いることで、選択した手の評価値を算出できる。

2.3 強化学習

機械学習の1つであり、ある環境内におけるエージェントが、現在の状態を観測し、取るべき行動を決定する問題を扱う。

2.4 ニューラルネットワーク

ニューラルネットワークとは、人間の脳神経系のニューロンを数理モデル化したものの組み合わせである。正解情報を入力する教師あり学習と、それを入力しない教師なし学習に分類される。画像や統計などの多次元データで線形分離不可能な問題に対して、従来の方法よりも比較的小さい計算量で良好な解が得られることが多い。

2.4.1 畳み込みニューラルネットワーク

畳み込みニューラルネットワークとは、深い層を持つニューラルネットワークであり、画像認識など二次元情報の分野で優れた性能を持つ。

今回扱うオセロでは、盤面上を白、黒、空白で分けられた8*8の画像データ、すなわち二次元情報として扱うことが可能なので、畳み込みニューラルネットワークが適している。

3 関連研究 (静的評価関数)

‘TDMC(λ)に基づく評価関数の調整’よりオセロゲーム進行度別の静的評価関数を用意した。この研究はTD学習に各非終端局面の勝率を導入させた強化学習法であるTDMC(λ)によって、評価関数のパラメータ調整を行ったものである。

4 実験・検証

実験方法はプレイヤーとCPUモデルがオセロにて対戦を行う。CPUモデルの動作決定には、プレイヤーの練度を基準に行う。プレイヤーの練度は $\alpha\beta$ 法、静的評価関数を

用い測定し、数値化する。一定回数以上対戦を繰り返し、プレイヤーの勝率が4割～6割となったとき、検証成功とする。

ゲームルールにおいては基本的なオセロと変わらないが、CPUは常に後攻という制約がある。

プレイヤーの練度測定には $\alpha\beta$ 法、静的評価関数を用い、算出された評価値を基準とする。

CPUモデルの動作内容は、打てる手全ての評価値を算出し、プレイヤーの練度を基準として最も近い値を選択する。

検証するにあたって実際に人間をプレイヤーとして本研究に参加させた場合、長時間の検証により疲弊し、ゲームプレイにおいて整合性が取れなくなることが予想される。そこでプレイヤーを模したCPUモデルを作成した。プレイヤーは戦術上、ケアレスミスなどにより常に最善手を選択するわけではないという考えから作成されている。動作内容は、打てる手全ての評価値を算出し、あらかじめ設定された範囲の中から乱数で値を選択するというものである。

実験結果は下記の表である。表内で用いられている各名称について記す。

“上限”、“下限”... プレイヤーを模したCPUモデルにおいて、あらかじめ設定された精度の範囲。数字が増えるごとに精度が落ちる。

“プレイヤー”、“CPUモデル”、“引き分け”... それぞれプレイヤー(黒)、CPUモデルの勝利数、引き分け回数を表す。

表1 実験結果

上限	下限	プレイヤー	CPUモデル	引き分け
0	0	27勝	20勝	3回
0	1	27勝	21勝	2回
0	2	28勝	22勝	0回
0	3	25勝	24勝	1回
1	2	23勝	26勝	1回
1	3	27勝	22勝	1回
1	4	22勝	25勝	3回
2	3	24勝	22勝	4回
2	4	31勝	17勝	2回

5 まとめ

実験結果より目的は達成された。今回の研究の目的は、プレイヤーごとの練度に合わせた、ほどほどのCPUモデルを作成することであった。 $\alpha\beta$ 法を用い、適切なCPUモデルを作成することで、当初の目的は達成された。しかし今回作成したCPUモデルは、プレイヤーの直前の手のみを手決定の要素に採用しているだけであり、真にプレイヤーの練度に合わせた難易度を実現できているとは言い難い。本来はCNNを用い、過去のプレイヤーの手などを説明変数とすることで、適切な難易度設定を行うことが目的だった。しかし今回は実装に至らなかった。そこでCNNを用いた、プレイヤーの練度に合わせて手決定を行えるCPUモデル

の実装案を提案する。

5.1 CNNを利用したCPUモデルの実装案

CNNを用いた場合の説明変数、目的変数について提案する。

説明変数:プレイヤーの手の良し悪し P 、手の決定までにかかった時間 T 、勝敗 R

目的変数:CPUモデルの手の決定 H

これらの変数より、(3)式を提案する。

$$H = W_0 + W_1(P_1 + T_1 + R_1) + W_2(P_2 + T_2 + R_2) + \dots + W_n(P_n + T_n + R_n) \quad (1)$$

5.1.1 説明変数

・プレイヤーの手の良し悪し

プレイヤーの手がどれほど有効かを $\alpha\beta$ 法、静的評価関数を用い数値化する。得た数値を元に、CPUモデルは最適に近い数値を得る手を決定する。

・プレイヤーの手の決定までにかかった時間、総試合時間

プレイヤーが手決定までに要した時間、その手の評価値を随時計測し、CPUモデルはそれに見合った最適手を選択する。

・勝敗、勝率

あらかじめ設定された試合数をこなした後、求められた結果を出しているかを検証する。

5.2 目的変数

説明変数を用い、CPUモデルの手を決定する。

これらが実装された場合、CPUモデルはプレイヤーの直前手のみを評価対象にするのではなく、プレイヤーの過去の練度などを評価に加えられるようになると考えられる。またプレイヤーの練度から相対的に見た、CPUモデルの難易度設定なども説明変数として扱うことができれば、更にプレイヤーの求めるCPUモデルに近づくことが予想される。

6 参考文献

- [1] 大崎 泰寛, 柴原 一友, 但馬 康宏, 小谷 善行: "TDMCAに基づく評価関数の調整", 情報処理学会シンポジウム論文集 (情報処理学会ワークショップ論文集), 2008 巻11号 pp.73-79 (2008/10/31).
- [2] 松本 渉, 小林 康幸: "UCT探索における局面評価関数の使用方法と性能評価", The 18th Game Programming Workshop 2013
- [3] shiracamus: 「プログラミング初心者がpythonの勉強がてら、オセロAIを作ってみた」をリファクタリングさせていただいた,
https://qiita.com/shiracamus/items/f03127819fff0f1a4349#_reference-40542ad712767f14034b