機械学習を利用した対局型ゲームのレーティング手法
A Machine Learning-Based Method for Rating Adversarial Games

島貫　凌世
Ryosei Shimanuki
法政大学情報科学部コンピュータ科学科
E-mail: ryosei.shimanuki.3j@stu.hosei.ac.jp

Abstract

Blockchain-based games that enable players to engage in peer-to-peer item transactions have recently gained remarkable popularity. Since these games transfer in-game assets based on match results, appropriate player matching according to their skill levels is becoming more important. However, the traditional Elo rating system, which is widely used to rank player abilities, faces significant challenges. One primary problem is "smurfing," where experienced players create new accounts to play against less skilled opponents. This paper proposes a new rating system called ML_Elo that is tailored for the game of Othello. By harnessing the power of machine learning, this system aims to more accurately predict a player's skill and appropriately adjust the fluctuation value in his/her rating. This promotes fairer in-game player matching and mitigates the negative impacts of smurfing and other gaming malpractices. The proposed method is applied to a virtual Othello environment using AI players. The creation of the AI players utilizes a learning model based on neural networks trained on actual game records. The experiment evaluates the transition of individual players' rating values and the distribution of overall rating values, comparing the proposed method with the standard method.

1. はじめに

近年，ブロックチェーンゲームに代表される，対戦結果によって資産としてのアイテムが動くゲームが登場している．このようなゲームでは，適切な実力を持つプレイヤー同士がマッチングされることの重要性が高まっていくことが予想される．そこで重要となるのがゲーム環境のレーティング手法である．Elo レーティング（以下，ER）に代表される従来のレーティング手法では，オンライン対戦ゲームで既に一定以上の実力を持つプレイヤーが新規のアカウントを作成することで実際の実力を偽ったり，実力的に適切でない相手とマッチングしたりすることが可能である．これらは「初狩り」と呼ばれる行為に代表され，ゲームシーンでは問題視され続けている．

本論文では，機械学習によってプレイヤーのレーティング値を適切な値により早く収束させることを目指す新型 ER を提案する．ER の問題点は，レーティング値の変動に大きく関わる値Kが多くの場合定数となっている点，その値が人為的に決定されている点である．本研究では，具体的な対局型ゲームとしてオセロを選択し，レーティングにおいて重要な特徴量を調査・学習し実力のあるプレイヤーのレーティング値を正確かつ迅速に収束させることを目指す．

2. 関連研究

これまでの関連研究では，棋譜解析を用いたプレイヤーの実力推定が多く行われてきた．馬場ら [1]は，将棋クエストを対象としてレーティング値 1100 から 100 刻みに 2099 までの局面を集計し，棋譜解析ソフト Bonanza が算出するプレイヤーの一手の平均損失とレーティング値の相関関係を調査した．結果，評価値の絶対値が少ないほど少ない局面数でプレイヤーのレーティング推定ができる可能性が示唆されたものの，評価値の閾値を 300 としたときに必要な分析局面を 50 局用意するには人間同士の対局で 14 局程度の対戦が必要であり，収集の負担が大きいという問題点が指摘された．高津ら [2]は，現在の ER における問題点を指摘している．藤井聡太棋士に注目し，29 連勝を果たした当時の将棋連盟棋士別成績一覧からプロ棋士のレーティング値の分布を調査して，各レーティング帯の 29 連勝する確率を調査した．結果，その確率が現実的なものとなってくるのはレーティング値 2100 以上の 21.9%であり，これは当時の藤井棋士のレーティング値 1800 を大幅に上回っていた．高津らは ER の問題点として，初期値の設定が正しいのか，ER では短期間の大きな実力の向上を十分にレーティング値に反映できないといった点を挙げた．また，オセロについては Takizawa [3] によってその弱解決がなされた．

3. 準備

3.1. Elo レーティング

Elo レーティングは Arpad Elo によって考案されたレーティングシステムであり，あるプレイヤーのレーティング値Rは環境における平均的なレーティング値をR_0，勝利確率W，敗北確率Lとして$R = (400 \log_{10} W/L) + R_0$と定義される．プレイヤー$A, B$のレーティング値をそれぞれ$R_A, R_B$として$A$の期待勝率$E$は$E = 1/(1 + 10^{(R_B - R_A)/400})$，$A$の新たなレーティング値$R'$は試合結果を$S$，変動因子を$K$として$R' = R_A + K(S - E)$と定義される．$S$は勝利，引き分け，敗北で，それぞれ 1，0.5，0 の値が割り当てら

Supervisor: Prof. Hiroshi Hosobe

れる．AがBに勝利する確率W_{AB}はそれぞれのレーティング値から$W_{AB} = 1/(1 + 10^{\frac{R_b - R_a}{400}})$と算出される．

3.2. オセロクエスト

オセロクエスト(以下，OQ)は棚瀬寧が開発し，Mindwalk 社が運営するクエストシリーズの一つであり，ER に準じたレーティングシステムが採用されている．対局数が少ないうちは「仮レート」状態としてレーティングの変動幅が大きくなり，表示上のレーティングも真のレーティングより低くなる．長期間対戦していない場合も同様に仮レート状態となる．

4. 提案手法

本研究では対局型ゲームにおけるプレイヤーの実力を正確に反映し，レーティング値の収束を早めることを目的として，ML_Elo レーティングを提案する．

4.1. 手法

試合結果の棋譜から実力の推測に関わる特徴量を抽出し，そこからレーティング値を予測し新たな変動因子K'を設定する．レーティング値の予測には実際の棋譜の特徴量と参加プレイヤーのレーティング値をニューラルネットワーク（以下，NN）で学習させた学習モデルを用いる．このとき，変動因子K'の式は以下のように表される．

$$K' = (R_e - R_A)/(S - E)$$

提案手法によるレーティング値は定数αを用いて以下のように表される．

$$R'_m = R_A + \alpha K'(S - E)$$

係数αの値については実験の中で検討する必要があるが，基本的に学習モデルのレーティング値の予測精度に基づいて$0 < \alpha < 1.0$の範囲で決定する．また，レーティング値の収束を早める必要があるプレイヤーの判別については ER 値の差と連勝数から決定する．ある試合に参加したプレイヤーA のレーティング値をR_A，対戦相手のレーティング値をR_B，提案手法の適用条件となるレーティング差をR'，連勝数をW^*として，試合結果から，勝利した相手とのレーティング差がR'以上，または連勝数W^*以上のプレイヤーに提案手法によるレーティング値補正を行う．

R'，W^*の値については ER の定義に基づいて決定する．R'について，プレイヤーA，Bの間で$R_A > R_B$とする．このとき ER の定義から$R_A - R_B = 400$の場合，$W_{AB} = 90.91(\%)$である．次に連勝確率について ER の定義から，仮に特定のプレイヤーが常に自身とのレーティング差が± 100以内のプレイヤーと対戦し続けた場合の平均連勝確率は表 1 のようになる．8 連勝する確率は0.72%であり，現実的な値ではない．本研究では以上を踏まえ，$R' = 400$，$W^* = 8$とする．また，条件を満たしたプレイヤーの対戦相手のレーティングには従来手法を適用するため変動因子補正の影響を受けることはない．

表 1 レーティング差 100 以内の連勝確率

連勝数	3	4	5	6	7	8
連勝確率(%)	13.5	7.27	3.99	2.23	1.26	0.72

4.2. レーティング値予測モデル

提案手法のR'_mは棋譜の特徴量からプレイヤーのレーティング値を予測する学習モデルから算出する．本研究では学習モデルの作成に利用する特徴量としてプレイヤーのレーティング値（R_p），勝敗（S），隅を取った数（C），平均総開放度（O_{avg}），対戦相手とのレーティング値の差（R'）を採用する．まず，勝敗（V）はそのプレイヤーの試合の結果を表し，次の式で表される．

$$V = \begin{cases} 1 & (\text{勝利}) \\ 0.5 & (\text{引き分け}) \\ 0 & (\text{敗北}) \end{cases}$$

平均開放度（O_{avg}）はオセロゲームの盤面における各石に隣接する空きマスの数の平均を表し，Nを盤面上の石の総数，O_iをi番目の石に隣接する空きマスの数としたとき，次の式で表される．

$$O_{\text{avg}} = \frac{1}{N}\sum_{i=1}^{N} O_i$$

平均総開放度（$O_{\text{avg\_t}}$）は複数の盤面にわたる開放度の総和を平均した値であり，盤面の数をM，試合中の手数をTとしたとき次の式で表される．

$$O_{\text{avg\_t}} = \frac{\sum_{i=1}^{M} O_i}{T}$$

レーティング差（R'）はプレイヤーのレーティング値をR_p，対戦相手のレーティング値をR_oとしたとき，次の式で表される．

$$R' = R_p - R_o$$

学習モデルを，特徴量ベクトルを入力としてレーティング予測値R_eを出力する関数$G(v)$と定義すると，R_eは以下の式で定義される．

$$R_e = G(V, C, O_{\text{avg\_t}}, R')$$

5. 実装

全ての実装は Python で行っている．

5.1. AI の作成

AI の実装では NN を用いて実際の OQ プレイヤーの棋譜（http://questgames.net/reversi から収集）を学習した打ち手予測モデルを利用した．作成したオセロ AI はランダムな手を打つランダム AI，現在の盤面を入力として，学習モデルから最善手を予測する AI（以下，予測 AI），αβ法と学習モデルを組み合わせた AI（以下，on_model αβ AI）の 3 種類である．on_model αβ AI では従来のαβ法のアルゴリズムの探索ノードを学習モデルの予測からソートしている．この学習モデルは次手が 64 マスのどれに属するかを全てのマスについて確率で出力する．打ち手予測モデルの作成では，AlphaGo [4]を参考として，次手の予測を，棋譜を入力として次手が 64 個のどのマス目に属すのかを分類する 64 クラス分類問題として定義した．オプティマイザは確率的勾配降下法，損失関数はカテゴリカルクロスエントロピーを利用した．トレーニングでは OQ 上位 4%のプレイヤーの最新の棋譜約 1300 試合分を入力として 20 Epoch ほど学習させた．AlphaGo では Accuracy ≅ 0.57 だったことから，本研究では Accuracy ＝0.60付近で学習を打ち切った．レーティング

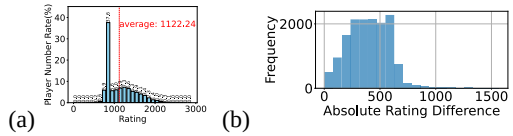
値予測モデルの学習では，損失関数に平均二乗誤差（以下，MSE）を採用し，約 1500 試合分のデータを 1000 Epoch ほど学習させた．学習の結果，MSE = 10416，$\sqrt{\mathrm{MSE}} \cong 102.05$であった．

5.2. シミュレーション環境の作成

提案手法を評価するシミュレーションモデルとして仮想的な OQ 環境（以下，VIOQ）を実装した．実装したシステムは従来の ER によるレーティングシステム，マッチングシステム，マッチングした AI 同士の複数の試合を並列して行うシステム，提案手法の 4 つである．VIOQ では最初に 1000 のプレイヤーのデータを初期化する．各プレイヤーは ID，レーティング値，AI タイプ，ランダム着手確率，オンラインフラグ，連勝数の 6 種類のパラメータを保持しており，全てのプレイヤーのデータは 1 つの CSV ファイル内で管理されている．時間単位を現在マッチングしている試合が全て終了するまでと定義し，1 Epoch と表現する．1 Epoch ごとに全体の 2.5%のプレイヤーがオンラインとなり，オンラインプレイヤーの中から許容レーティング差以内のプレイヤー同士をマッチングし並列で試合を行う設計とした．1 Epoch が終了した後，試合結果を全プレイヤーのデータベースに反映し，次のマッチングを開始する．レーティングシステムについて提案手法との違いを明確にするため，従来手法を適用する場合は変動因子Kを常に固定値で設定している．

6. オセロクエストの分析

OQ の全プレイヤーの分布を調査した結果を図 1 に示す．図 1(a)は 0 を始めとして 100 刻みのレーティング帯毎のプレイヤー数の全体に対する割合を表している．全プレイヤーの中で最も分布の割合が高かったのはレーティング帯 800～900 で全体の37.6%だった．OQ の初期レーティング値は800で設定されているため，アカウント作成後の対戦回数が著しく少ない，または勝利数を稼げていないプレイヤーが半数近くとなっていることが考えられる．図 1(b)は最上位のプレイヤーがマッチングした全期間の対戦相手とのレーティング値の差の絶対値の分布を示している．調査の結果，最上位のプレイヤーを除けば，基本的にマッチングするプレイヤーとのレーティング値の差R'_Mは$-600 \leq R'_M \leq 600$の範囲に収まっていることがわかった．



(a) (b)

図 1　(a)OQ 全体のレーティング値の分布，(b)レーティング値 2411 のマッチング相手の分布

7. 実験

VIOQ でシミュレーションを行い，どのパラメータがレーティング値の効率的な収束に繋がるかを検証する．

7.1. シミュレーションモデル

VIOQ で OQ の棋譜を学習させた AI プレイヤーによって実際の環境を再現する．シミュレーションではランダム着手確率によって，同一の AI を利用するプレイヤーでも異なるプレイヤーとして評価する．環境の初期化時に全体の2.5%のプレイヤーがオンラインとなり，自動的にマッチングされる．以降は 1 Epoch が終了する毎に全試合の結果が反映され 3 Epoch 毎にオンラインプレイヤーが切り替わり次のマッチングが実行される．以降は同一の操作が終了 Epoch まで繰り返される．

表 2　環境上に実装した AI プレイヤーの分布

AI の種類	全体の割合	初期レーティング値
ランダム AI	50%	1100
予測 AI	25%	1100
on_model αβ AI	25%	1100

7.2. 実験 1

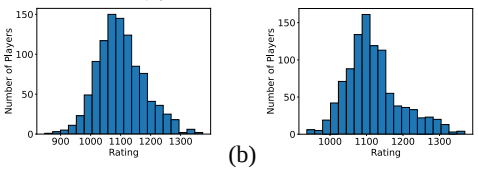
実験 1 では，環境のレーティング値の分布・実力のあるプレイヤーのレーティング値を効率的に収束させるために最適なレーティング初期値，定数αを求める．また，環境の実行時に初期化される 1000 プレイヤーに加え，各パラメータを事前に指定した 5 プレイヤーを初期化する．指定したプレイヤーのレーティング値の推移を初期化時から追跡し，従来手法と比較する．OQ の調査結果から，実験の初期設定では従来手法の変動因子$K = 32$，$\alpha = 0.5$，マッチングを許容するレーティング値の差は600とする．

7.3. 実験 2

実験 2 では，全体のレーティング値の分布が十分に成熟したと判断できる Epoch で新たなプレイヤーを追加し，追加したプレイヤーのレーティング値の推移を従来手と提案手法で比較する．

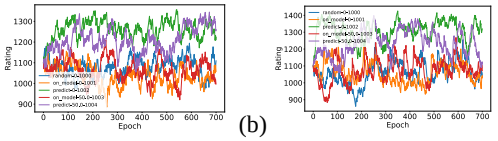
7.4. 実験 1 の結果

従来手法，提案手法をそれぞれ適用した環境（以下，環境 B，環境 P）の一定 Epoch 経過後のレーティング値の分布を図 2 に，指定したプレイヤーのレーティング値の推移を図 3 に示す．環境 B，P に全体的なレーティング値Rの分布は$900 \leq R \leq 1400$の範囲に収まっていた．また，指定したプレイヤーのレーティング値の推移を比較すると，各環境で予測 AI プレイヤーが全 Epoch を通して他プレイヤーよりも高い傾向にあった．このことから，現環境では予測 AI を採用したプレイヤーが最も実力のあるプレイヤーであると判断できる．指定プレイヤーのレーティング値の傾向に各環境で大きな差はなかった．



(a) (b)

図 2　(a)環境 B（3200 Epoch 経過時），(b)環境 P（700Epoch 経過時）のレーティング値の分布



(a) (b)

図 3　(a)環境 B，(b)環境 P の指定プレイヤーのレーティング値推移（700Epoch 経過時）

7.5. 実験 2 の結果

各環境の 700 Epoch 経過時に新規プレイヤーを参入させた場合の，予測 AI プレイヤーのレーティング値推移を比較した結果を図 4 に示す．各環境の全体的なレーティング値の傾向に大きな差は出なかった．実験 1 から最も実力のあるプレイヤーと結論された予測 AI プレイヤーについては，環境 P で取るレーティング値は環境 B で取る値と比較して殆どの Epoch で高い値をとる傾向にあった．

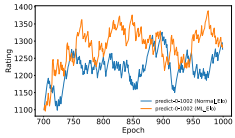
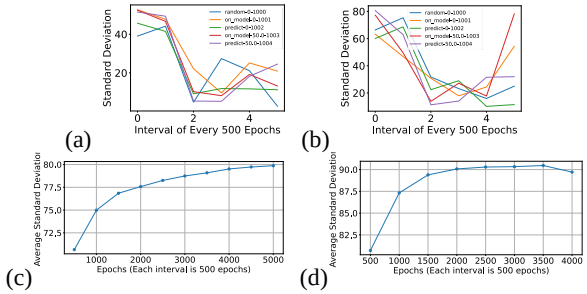


図 4　予測 AI の従来手法・提案手法によるレーティング値推移の比較

7.6. 標準偏差による評価

各環境の個別プレイヤーのレーティング値の 500 Epoch 刻みの標準偏差の推移，全体のレーティング値の分布の標準偏差の推移を図 5 に示す．図 5(a)，(b)から個別プレイヤーに関して，提案手法では on_model αβ AI タイプのプレイヤーについてのみ従来手法と比較してレーティング値の推移が不安定になるといった結果となった．一方で，図 5(c)，(d)から，提案手法では従来手法と比較して全体のレーティング値の分布の変化が早く安定，収束する傾向にあることがわかった．



(a) (b)

(c) (d)

図 5　(a)環境 B(b)環境 P の個別プレイヤーの 500Epoch 刻みの標準偏差の推移(c)環境 B(d)環境 P の全体のレーティング値の分布の標準偏差の推移

8. 議論

実験 1 の結果から，環境 P のレーティング値の分布について，3200 Epoch 経過時の環境 B レーティング値 1200 以上のプレイヤーの分布の特徴が早期に現れていることから，提案手法によって全体の分布の先取りができていると考察する．また，個別プレイヤーについては実験 2 の結果，図 3 の指定プレイヤーのレーティング値の推移から，現環境で最も実力のある AI は予測 AI と判断できる．この予測 AI タイプのプレイヤーについては図 4 から，提案手法によってより適正なレーティング値に早期に補正できていると考察される．これは，不適切な値であれば敗北を繰り返すことで値が下がると考えられるためである．結果として，提案手法による個別プレイヤーのレーティング値推移と全体の分布推移の安定性について図 5 から全体の分布の安定性は向上した一方で，中間層の実力のプレイヤーのレーティング値推移の安定性が低下したことが考えられる．これは提案手法によるレーティング値補正の適用条件として連勝数を採用していることが影響していると考えられるため，今後は連勝数以外の適用条件を検討する必要がある．

また，提案手法では$\alpha = 0.3$としているためレーティング値予測モデルの予測結果を効率的に活用しているとは言えない．このことからもレーティング値予測モデルの学習精度，または値の予測に採用した特徴量が不適切であった可能性が高いと言える．プレイヤーのレーティング値の分布についても，OQ では 2000 の幅があるのに対し，環境 B，Q 共に 500 程度しかなかった．以上から，実際の OQ 環境に提案手法を適用した場合の傾向がどのようになるかの予測が難しい．加えて，今回の実装ではプレイヤーの成長を再現できていない．今後は学習モデルの精度，OQ 環境の再現度，加えて中間層のプレイヤーのレーティング値推移の安定性への影響を改善する必要がある．

9. おわりに

本論文では，対局型ゲーム，特にオセロについて実力のあるプレイヤーを早期に適切なレーティング値へと収束させるレーティング手法を提案した．プレイヤーのレーティング値が収束したかの議論は難しく，シミュレーション結果からの推測はできたものの収束したかどうかの結論を出すことはできなかった．学習モデルの向上，環境の改善，レーティング値の収束をどのように議論・定義するかが今後の課題である．

文　献

[1] 馬場匠, 伊藤毅志, "少ない棋譜からの将棋プレイヤ棋力推定手法の研究," ゲームプログラミングワークショップ論文集, no. 1, pp. 183-186, 2018.

[2] 高津和紀, 高田宗樹, 平田隆幸, "将棋の最年少プロ棋士藤井聡太の強さを測る：レーティングによる評価と問題点," 福井大学大学院工学研究科研究報告書, vol. 67, 2018.

[3] H. Takizawa, "OTHELLO IS SOLVED," 2023. [Online]. Available: https://arxiv.org/abs/2310.19387.

[4] S. David, H. Aja, M. Chris J, A. G, S. Laurent, G. v. d. Driessche, S. Julian, A. Ioannis, P. Veda, L. Marc, D. Sander, G. Dominik, N. John, K. Nal, S. Ilya, G. Thore and H. Demis, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 27, pp. 484-489, 2016.