

習熟度に基づく最適戦略の自動選択に関する研究

—P-to-C 対戦型ゲームを例にして—

M2021SE011 竹内大輔

指導教員：野呂昌満

1 はじめに

Hooshyar らは、2018 年時点で、ゲーム AI 設計において以下の問題があると述べている [1].

- 習熟変化による適応方法が未確立
- 異なるゲーム間での習熟測定が困難

両者を整理すると、以下の問題であると解釈できる.

- プレイヤーの習熟が測定できない
- 習熟に基づく最適戦略選択ができない

プレイヤーの習熟を測定して、ゲーム AI の最適戦略を自動選択することは、ゲームの質を向上させることにつながる。単一のゲームにおいて習熟度を定式化することは困難であるが、複数のゲームにおいてはさらに困難になる。同様に最適戦略を自動選択させることも困難である。これらの問題は、現状、十分に議論されていない。

本研究の目的は、対戦型ゲームにおいて、習熟度に基づく戦略の自動選択を発見的に行う方法について考察することである。すなわち、異なるゲームのプレイヤーに対して適切に習熟測定を行い、習熟に基づいた最適戦略の自動選択を行う方法を考察する。異なるゲームにおける習熟度の定量化方法を提案する。さらに、習熟度に基づく適切な対戦戦略を自動選択するアーキテクチャを定義する。

技術課題を以下に示す。

RQ1 習熟度の発見的な測定方法の提案

RQ2 習熟測定と自動選択を行う仕組みの提案とアーキテクチャ定義

RQ3 着想の妥当性検証

これらを解決することで、対戦型ゲームで意欲を刺激するゲーム AI の設計が可能になる。

2 関連技術と関連研究

本研究では、表現学習を利用する。表現学習は、多様な入力を低次元の特徴量に変換する技術である。表現学習を用いて、異なるゲームをプレイするプレイヤーの入力を特徴量に変換することで、異なるゲームの習熟測定が可能になると考えた。

関連研究として、水谷らは、画像改ざんの検出方法として、改ざんの方式を特定するアーキテクチャを設計している [8]。特定の画像改ざんを検出するニューラルネットワークを予め複数用意しておき、どれを使用するかを別のニューラルネットワークで決定する。改ざんされた画像に対して、どのような改ざんがされているかを、複数のニューラルネットワークを用いて多数決を取る形で予測する。予測結果に基づいて、改ざん内容の検出に用いるニューラルネットワークを選択することで、高い精度

で画像の改ざんを検出できる。改ざん方法を特定して適切なニューラルネットワークを選択する方法は、本研究における習熟測定に応用できると考えた。以降、ニューラルネットワークのことを NN と表記する。

3 技術課題解決のアプローチ

技術課題を解決するためのアプローチとして、機械学習を用いた習熟測定と戦略選択を行う。習熟度の定量化を行うために、表現学習を用いて習熟度をベクトルに変換する。表現学習を用いて、異なるゲームの入力を特徴量に変換することで、入力内容に基づく習熟を表現できると考えた。すなわち、異なるゲームでの習熟測定が可能になると考えた。

同様の理由で、習熟測定と戦略選択にも機械学習を用いる。習熟を示す特徴量を用いて機械学習を行い、適切に習熟を定量化する。定量化した習熟度を用いて機械学習を行い、習熟に基づく最適戦略の自動選択を行う機構を実現する。

4 アーキテクチャ定義

4.1 ゲーム AI の参照アーキテクチャ

習熟測定と自動選択を行う仕組みを定義するために、アーキテクチャを定義する。図 1 は、一般的なゲーム AI の参照アーキテクチャである。この構造では、コンテキストに基づいてビヘイビアアクティベータが作用して、AI コンポーネントの動作を決定する仕組みとなっている。習熟度がコンテキスト、最適戦略の自動選択を行うコンポーネントがビヘイビアアクティベータに該当する。

図 1 において、コンポーネントのモデルが NN のクローニングであると表現した。これは、各コンポーネントが NN にクローニングされて運用されることを示している。本研究では、システムにおける各工程を NN によって実現する。

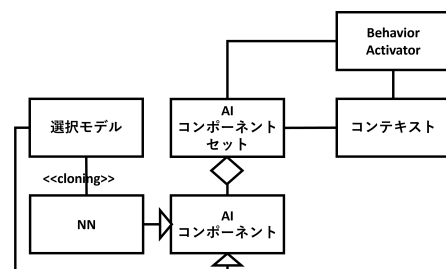


図 1 ゲーム AI の参照アーキテクチャ

4.2 概念アーキテクチャ

図2は、習熟度に基づいて戦略選択を行うことを示した概念アーキテクチャである。習熟度をコンテキストとして、習熟度に基づいて行動選択器というコンポーネントが作用する構造を示している。行動選択器については、4.4節で説明する。

図1を詳細化して、コンテキストである習熟度によって行動の切り替えが行われることと、行動選択器によって行動が選択されることを明確化した。すなわち、行動選択器が習熟度に基づいて戦略自動選択を行うという構造を明確化した。

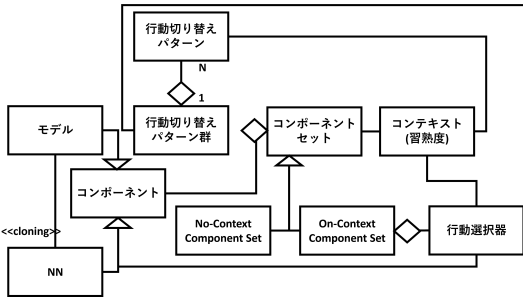


図2 概念アーキテクチャ

図2では、コンテキスト指向に基づく設計を行うために、用いるコンポーネントをコンテキストに関与するかしないかで分類している。On-Context Component Setは、コンテキストを決定する、あるいはコンテキストによって挙動が変化するという、コンテキストに直接的な関係があるコンポーネントの集合である。No-Context Component Setは、On-Context Component Setに該当しない、すなわちコンテキストに直接的な関係がないコンポーネントの集合である。

4.3 具象アーキテクチャ

図3は、習熟度をコンテキストとして定義した具象アーキテクチャである。本研究では、AIコンポーネントに該当するものを4つ用いることで、習熟測定と最適戦略選択を行う。各コンポーネントの詳細は4.4節で説明する。

図3では、4つのコンポーネントのうち、3つがNNのクローニングであると表現した。これは、各コンポーネントがNNにクローニングされて運用されることを示している。

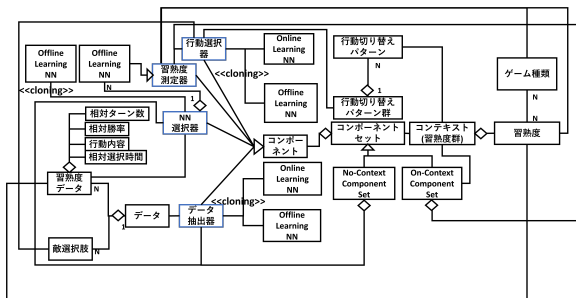


図3 具象アーキテクチャ

4.4 各コンポーネントの概要

図4は、前述した各コンポーネントとFront Endの静的構造を示した図である。各コンポーネント同士でやりとりを行うことで、システムを運用する。本節では、各コンポーネントの仕組みと役割について述べる。

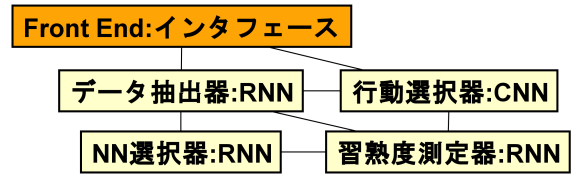


図4 アーキテクチャの静的構造

4.4.1 データ抽出器

データ抽出器は、習熟に影響するデータを抽出するコンポーネントである。このコンポーネントは、3節で述べた表現学習を行うコンポーネントである。

データ抽出器は、プレイヤーの入力とゲーム内の値を入力して、以下に該当するデータに対して表現学習する。

- 習熟度データ
- 敵選択肢

習熟度データは、ゲーム内の値のうち習熟の影響を受けるデータの総称である。習熟度データを抽出する理由は、習熟による値の変化から習熟度測定を行うからである。本研究では、習熟を発見的に測定するために、データ抽出器を用いて習熟度データの表現学習を行う。本研究の習熟度データは、以下の4つである。

- 相対ターン数：ゲームプレイにおける経過したターンの数
- 相対選択時間：プレイヤーがゲームの行動決定に要した時間
- 相対勝率：プレイヤーのゲームプレイにおける勝利・敗北数の比率
- 行動内容：プレイヤーがゲームの行動決定において選択した行動の内容

敵選択肢は、ゲーム内で敵AIの戦略として選択される行動選択肢群である。敵選択肢を抽出する理由は、ゲームによって用意されている選択肢が異なり、敵AIの選択を決定的に定式化できないからである。

4.4.2 NN選択器

NN選択器は、習熟測定に用いる測定方法を決定するコンポーネントである。習熟度の定量化には、習熟度データの標準化以外にも、習熟の変化傾向の計測が必要である。データ抽出器による表現学習だけでは習熟度を定量化できない。本研究では、習熟変化傾向を計測するために、習熟測定方法を切り替える。習熟測定方法として用いるNNを切り替えて、習熟度の定量化方法を選択する。

NN選択器は、習熟度データを入力して、習熟測定に適した方法がどれであるかを出力する。習熟度データを用

いて、どの習熟測定方法が適しているかを予測して選択することで、習熟度の適切な定量化ができると考えた。

4.4.3 習熟度測定器

習熟度測定器は、プレイヤーの習熟度合いを習熟度に定量化するコンポーネントである。これまで述べた、習熟測定を行う仕組みがこのコンポーネントに該当する。習熟度は、本研究における最適戦略自動選択のコンテキストとなるので、習熟度の測定精度が本研究の根幹となる。精度の高い戦略選択を行うために、習熟度測定器はより精密な設計と実装が必要になる。

習熟度測定器は、データ抽出器の出力のうち習熟度データを入力して、プレイヤーの習熟度を定量化する。習熟度測定器に用いる NN は、NN 選択器で選択されたものである。本研究で述べる習熟測定方法は、習熟度測定器に用いる NN として実現するものとする。

4.4.4 行動選択器

行動選択器は、最終的に敵 AI が選択する行動を決定するコンポーネントである。これまでに述べた、最適戦略の自動選択を行う仕組みがこのコンポーネントに該当する。行動選択器では、ゲームの難易度がプレイヤーに適したものになるように、敵 AI の最適戦略を推定する。習熟度に基づいて行動選択内容を変化させることで、プレイヤーの習熟に適した対戦戦略の自動選択を実現する。

行動選択器は、習熟度と敵選択肢を入力して、習熟に適した選択肢を出力する。敵の行動は、データ抽出器から得た敵選択肢の中から選択される。これは、現在プレイ中のゲームで想定されていない行動内容が選択されないようにするためである。

4.5 各コンポーネントの学習と実装

本研究では、4.4 で述べた 4 つのコンポーネントにおいて、NN を用いた実装を想定している。異なるゲームにおいて習熟評価や最適戦略選択を行うのは、決定的な方法では実現できない。本研究では各コンポーネントにおいて機械学習を用いて、異なるゲームでの対応方法を発見的に予測・学習する。

本研究では、各コンポーネントにおいて、異なる方法を用いて学習・訓練および運用を行う。データ抽出器と行動選択器は、異なるゲームで異なる入出力が得られるので、運用前のオフライン学習と運用時のオンライン学習を併用して、異なるゲームに対応する。NN 選択器と習熟度測定器は、習熟の定量化にかかわるコンポーネントであるので、運用前にオフライン学習を行い、訓練を完了させたうえで運用を行う。

4.6 コンポーネント間の動的挙動と相互作用

本研究で提案する仕組みは、4 つのコンポーネント間でデータのやり取りを行うことで、データの表現学習と習熟測定、戦略の自動選択を行う。それぞれの工程は以下のとおりである。

1. ゲーム内の値から習熟度データと戦略選択に必要なデータを抽出

2. 習熟度データを用いて、習熟変化傾向を測定するのに適した習熟測定方法を選択
3. 習熟度データを用いてプレイヤーの習熟度を測定
4. プレイヤーの習熟度に適した敵の行動を決定

各コンポーネントでの入出力を、上記の工程に沿った順で行うことで、システムを運用する。

図 5 は、コンポーネント間のデータのやり取りと処理の順を示した図である。前述した 4 つのコンポーネントを図に示した順に用いて、習熟測定と最適戦略選択を行う。図 5 に示した通りにすることで、プレイヤーの習熟に適した対戦戦略の自動選択ができると考えた。

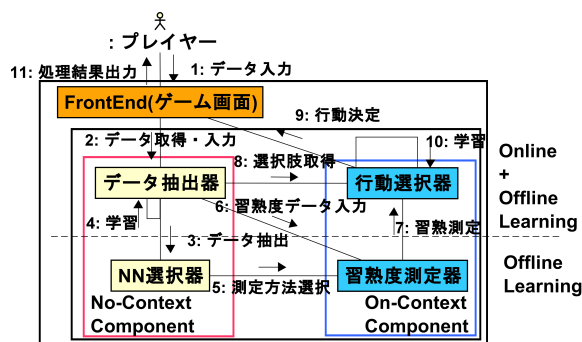


図 5 アーキテクチャの動的挙動

5 実験と評価

これまでの着想の妥当性を検証するために、提案アーキテクチャに基づくプロトタイプを実装して実験を行った。実際のゲームプレイを想定した実験とデータの収集を行い、機械学習技術の応用という観点から、アーキテクチャの妥当性を考察した。

図 6 は、NN 選択器がゲームに対する習熟測定方法を適切に判別していたかどうかの比率を示したグラフである。1 が正しく判別できていたとき、0 が間違っ判別していたときの比率である。図 6 の 1 と 0 の比率を計算したところ、複数回の実験において約 9 割が 1 であった。したがって、データ抽出器の表現学習とデータの抽出は、正しい習熟測定方法の判別に役立っていると考えた。

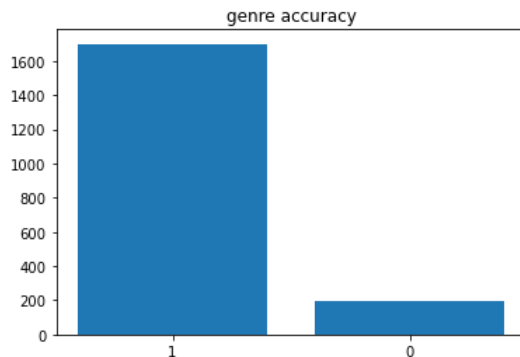


図 6 NN 選択器の正確性

図 7 は、実験全体における習熟度の推移である。全体を通してプレイヤーの習熟の評価が激しく上下しているため、習熟が上下していることを評価できていると考えた。したがって、プレイヤーの選択に応じて習熟の評価

をえることができていると考えた。

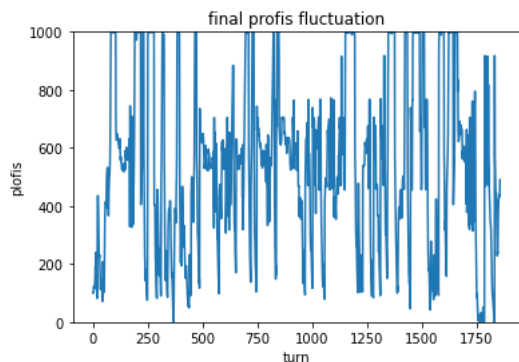


図 7 ゲーム全体の習熟推移

6 評価と考察

実験結果から、以下の観点について評価と考察を行う。

- RQ1 の解決方法についての評価
- RQ2 の解決方法についての評価
- RQ3 の解決方法についての評価
- 目的達成についての評価

RQ1 について、図 6 の結果から、ゲームに適した習熟測定方法の判別という点では、課題を解決している。機械学習を用いて表現学習する方法は、異なるゲームでの独立した習熟度測定に貢献できると考えた。

RQ2 について、データ抽出器および NN 選択器と関連する値や仕組みについては、図 6 の結果から、適切な予測と出力を行っている。習熟度測定器の仕組みは、図 7 の結果から、習熟の変化測定はできているので課題を解決したと考えた。習熟測定の精度について、グラフが滑らかでないので、精度改善の余地があると考えた。改善案として、水谷ら [8] の研究で用いられた、使用する NN を切り替える方法を行動選択器に適用すると、戦略選択の面から精度向上が見込めると考えた。

RQ3 について、実験の結果に基づく評価として、RQ1 と RQ2 を共に達成したと考えた。実験器の実装方法を改善することで、習熟精度の精度向上が見込めると考えた。

本研究の目的は、対戦型ゲームにおいて、習熟度に基づく戦略の自動選択を発見的に行う方法について考察することであった。本研究の目的は、技術課題の解決により達成した。したがって、対戦型ゲームで意欲を刺激するゲーム AI の設計が可能になった。

7 おわりに

本研究では、対戦型ゲームにおいて、習熟度に基づく戦略の自動選択を発見的に行う方法について考察することを目的とした。目的を達成するために、3つの技術課題を定義して、それぞれの解決方法について議論した。着想に基づく検証実験の結果、目的を達成した。

本研究の今後の展望は以下のとおりである。

- アーキテクチャの再設計
- プレイヤーの技能ごとの習熟度定義
- 協調フィルタリングを用いた設計と実装

参考文献

- [1] D. Hooshyar, M. Yousefi, and H. Lim, Korea University, “Data-Driven Approaches to Game Player Modeling: A Systematic Literature Review”, *ACM Computing Surveys*, Vol.50, No.6, Article 90, 2018.
- [2] Z. Huang, Q. Liu, and Y. Chen, “Learning or Forgetting? A Dynamic Approach for Tracking the Knowledge Proficiency of Students”, *ACM Transactions on Information Systems*, Vol. 38, No. 2, Article 19, 2020.
- [3] W. Min, et al, “Player Goal Recognition in Open-World Digital Games with Long Short-Term Memory Networks”, *IJCAI-16*, pp. 2590-2596, 2016.
- [4] J. Valls-Vargas, S. Ontanón, and J. Zhu, “Exploring player trace segmentation for dynamic play style prediction”, In *Proceedings of the 11th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, pp. 93-99, 2015.
- [5] 竹内大輔, “習熟度に基づき対戦戦略の動的変更を可能とするゲームエンジンアーキテクチャの設計”, 南山大学卒業論文要旨集, 2021.
- [6] 竹内大輔, 野呂昌満, 沢田篤史, “ゲーム対戦戦略をプレイヤー習熟度へ適応させる機械学習機構の設計”, 第 28 回ソフトウェア工学の基礎ワークショップ (FOSE2021), 2021.
- [7] A. Grover, J. Leskovec, “node2vec: Scalable Feature Learning for Networks”, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.855-864, 2016.
- [8] Akira Mizutani, Masami Noro and Atsushi Sawada, “Design of Software Architecture for Neural Network Cooperation: Case of Forgery Detection”, *APSEC-28*, pp.130-140, 2021.
- [9] Y. Yu. Dyulichева, A. O. Glazieva, “Game based learning with artificial intelligence and immersive technologies: an overview”, 4th Workshop for Young Scientists in Computer Science & Software Engineering, December 18, 2021.
- [10] P. Sajjadi, F. V. Broeckhoven and O. D. Troyer, “Dynamically Adaptive Educational Games: A New Perspective”, *Games for Training, Education, Health and Sports: 4th International Conference on Serious Games*, pp. 71-76, 2014.
- [11] 遠藤雅伸, 三上浩司, “フローゾーンを超えた動的難易度調整～イリンクスを楽しむ Dynamic Pressure Cycle Control 手法～”, *芸術科学会論文誌 Vol. 17*, No. 3, pp. 62-71, 2018..
- [12] B. Schweizer, H. Lowood, R. Guins, *Debugging Game History: A Critical Lexicon*, The MIT Press, 2016.