

線形補外とk近傍法を用いた 格闘ゲームにおける敵の位置と行動の予測

Prediction of the Opponent's Position and Action in a Fighting Video Game with the Linear Extrapolation and the k-Nearest Neighbor Method

浅山和宣 森山甲一 福井健一 沼尾正行
Kazuki Asayama Koichi Moriyama Ken-ichi Fukui Masayuki Numao

大阪大学 産業科学研究所
The Institute of Scientific and Industrial Research, Osaka University

In a fighting video game, it is difficult for an AI-controlled character (NPC) to defeat a very skilled human player under equal conditions; it is a challenging task to construct a strong NPC in the game. In this paper, we first investigate the relationship between the performance of an NPC and its rate of opponent perception. Then, we construct an NPC that predicts its opponent's future position and action, which simulates high-speed perception, with the linear extrapolation and the k-nearest neighbor method. The experiment shows that, out of six opponents, an NPC with two-frame prediction is stronger than the original one against all opponents, and that with 16-frame prediction becomes better against five opponents.

1. 序論

1940年代にデジタルコンピュータが完成して以来コンピュータの性能は爆発的に向上しており、さまざまな用途に利用されてきた。ゲームをコンピュータに行わせるという試みもその一つであり、ゲームを行うための人工知能アルゴリズム(以下AI)の研究も進んだことで、チェス、オセロ、将棋といったボードゲームの多くでAIは人間を凌駕しつつある。

一方リアルタイムコンピュータゲームについては、研究が広く行われてきた[1]ものの、AIの操作するプレイヤー(ノンプレイヤーキャラクター(NPC))は上手な人間プレイヤーに対等な条件ではまだ敵わない。リアルタイムコンピュータゲームでは、AIは複数のパラメータが刻々と変化する僅かな時間間に行動を選択しなければならず、限られた計算時間で膨大な計算を行わなくてはならないという点でボードゲームとは異なっている。そのため、人間の運動、認識を遙かに超えた動きをすることでAIにハンデが与えられている[2]という現状がある。

本研究では、リアルタイムコンピュータゲームの一種である格闘ゲームを対象とする。AIが人間の認識能力を越えた動きをせず、AIと人間プレイヤーが対等なキャラクターを操作することを条件とし、AIの操作するNPCが人間にとってより手強い対戦相手となり、多くの人間プレイヤーがNPCとの対戦を楽しめるようにすることを目的とする。

2. 格闘ゲーム FightingICE

格闘ゲームは人間やコンピュータがキャラクターを操作し徒手や武器を繰り出すことで対戦相手を打ち倒す事を目的としたリアルタイムコンピュータゲームの1ジャンルである。本章では、本研究で用いた格闘ゲーム FightingICE [3] のルールや仕様について説明する。以降、本論文ではキャラクターを制御し格闘ゲームのプレイを行うコンピュータのプログラムをエージェントと呼ぶこととする。

2.1 FightingICE とは

FightingICE は研究目的で開発された2D格闘ゲームで、エージェントをJava言語によって実装する。エージェントに独自の攻撃パターンや回避行動をとらせることだけでなく、対戦相手の動きから行動パターンを学習させる事も可能という自由度の高さを持つ。

2.2 ゲームのルール

FightingICE では1試合が3ラウンドで構成され、それぞれで得られた対戦スコア(2.5節で説明)の平均でその試合の勝敗を評価する。60秒を1ラウンドとし、時間が過ぎることで次のラウンドに移行する。また、このゲームでは1/60秒を単位時間としてフレームと呼び、エージェントは1フレーム毎に情報を得て、キャラクターに指示を送る。

2.3 HP

キャラクターにはHPというパラメータが設定されている。HPは対戦相手からどれだけダメージを受けたかを表すパラメータである。ダメージを受けると0から際限なく減少していき、ラウンドが変わる際に後述の対戦スコアの計算に利用されて0へとリセットされる。

2.4 ゲーム環境

エージェントは対戦中に情報として各キャラクターの位置と行動を取得し、これを元にどのような指示をするかを選択する。キャラクターの位置は、ゲームのウィンドウ左上を原点とし、水平右向きをx軸の正の向き、鉛直下向きをy軸の正の向きとした直交座標系で表される。

キャラクターの取り得る行動は、立っている(STAND)等の「何もしない」行動が3種類、DASH等の移動を表す行動が6種類、ガードを表す行動が3種類、空中から着地している(LANDING)等のある行動からの復帰を表す行動が12種類、THROW_A等の攻撃を表す行動が32種類で合計56種類となる。

2.5 対戦スコア

対戦スコアは、1ラウンド中に変動したHPから計算される。ラウンド終了時の対戦相手のHPを $oppHP$ 、自分のHP

連絡先: 浅山和宣, 大阪大学, 産業科学研究所沼尾研究室,
〒567-0047 大阪府茨木市美穂ヶ丘 8-1,
Tel: 06-6879-8426, Fax: 06-6879-8428,
E-mail: asayama@ai.sanken.osaka-u.ac.jp

を $myHP$ とすると、以下の計算式で求められる。

$$score = \frac{oppHP}{myHP + oppHP} \cdot 1000 \quad (1)$$

両方のプレイヤーの HP が等しい場合、つまり対戦が互角であった場合は対戦スコアが 500 になり、より多くのダメージを与えていた場合はそれに応じて 500 よりも大きな対戦スコアが得られる。対戦スコアは、どれだけダメージを受けずにダメージを与えられたかを表す指標である。

2.6 認識の遅れ

このゲームの大きな特徴の一つとして、エージェントの認識能力が制限されていることが挙げられる。エージェントは対戦相手の位置と行動（攻撃、防御など）の情報をゲームプログラムから取得するが、人間の認知能力の限界を模倣することを目的として、対戦相手の情報について 15 フレーム前（0.25 秒）のものしか知る事が出来ないという制約が与えられている。これにより、対戦相手の素早い攻撃を見てから、ガードやより早い攻撃でカウンターをするなどという人間離れした行動を NPC が取れないようになっている。

2.7 コンペティションと出場エージェント

FightingICE の開発元である立命館大学知能エンターテインメント研究室により格闘ゲーム AI のコンペティションが開催されている。2014 年のコンペティションに出場した各エージェントの特徴 [4] について簡単に説明する。

ATTeam2

対戦相手に届く攻撃を探索し行動を決定

DragonKing3C

対戦相手のモデリングを実行

PasanAI2

強化学習により有効な行動を予め獲得

Seal_Switch

遺伝的アルゴリズムで獲得した基準により行動パターンを変化

SejongFighter

予め決められた基準により適当な行動を実行

T3C

有限オートマトンを用いて作り込まれた行動を実行

3. 格闘ゲームと認識能力の関係

スポーツをする能力が高いとは現在の状況に応じて適切な行動を実行できる、つまり状況判断の能力に優れていると言い換えることが出来るだろう。トップレベルのボウリング選手を対象にした調査 [5] では、補欠選手に比べてスターティングメンバーの方が動体視力や焦点調節などの視覚機能において優れていたという結果が得られている。格闘ゲームは対戦相手の動きを瞬間的に把握して即座に有効な行動を取るという点でスポーツと酷似しており、格闘ゲームにおいてもスポーツと同様に認識能力が状況判断力に大きな影響を及ぼす要素であるということが示唆される。したがって、本章では格闘ゲームにおいてエージェントの認識能力が対戦相手に対して劣っている場合、対戦にどのような影響が出るのかを調査した結果を示す。

3.1 手法

格闘ゲーム FightingICE において、2014 年のコンペティション出場エージェントの一つである T3C を改変し、ゲームプログラムから得られる 15 フレーム前の情報をさらに遅らせることで対戦相手に比して認識が劣っている状態を意図的に作り出した（このエージェントを $delayedT3C$ と呼ぶ）。遅延フレーム数 d を 1 から 29 まで 1 刻みで、30 から 100 まで 10 刻みで変化させ、元の T3C を含む同コンペティション出場者である 6 種類のエージェントと各 10 試合ずつ対戦を繰り返し、対戦スコアを調べることで認識能力と格闘ゲームの能力との関係性を調査した。

3.2 実験と結果

対戦結果を代表して、 $delayedT3C$ と $PasanAI2$ の対戦結果を図 1 に示す。縦軸の値が大きければ大きいほど $delayedT3C$ の対戦スコアが高かったことを示す。赤で引かれた横線は、認識能力が遅れていない、つまり元の T3C が各対戦相手と戦ったときの対戦スコアである。これにより、認識の遅れが弱体化に繋がっていることが分かる。また、残り 5 つのエージェント中 4 つのエージェントについても認識の遅れが対戦スコアに悪影響を及ぼすことが分かった。

この結果から、逆に対戦相手よりも認識が速くなれば対戦スコアが改善されることが示唆される。しかしゲームのルール上、各エージェントの認識遅れは 15 フレームで一定である。そのため、認識を速くすることと同等の結果を得るために、対戦相手の未来の位置と情報を獲得可能な過去の情報から予測する手法を提案する。

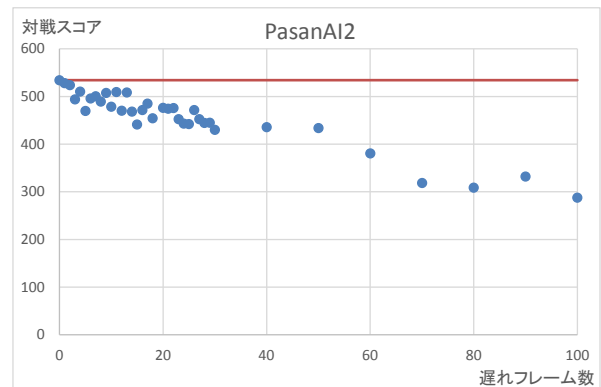


図 1: 得られる情報が遅れたときの $PasanAI2$ との対戦スコア

4. 対戦相手の位置と行動の予測

本章では、対戦相手の位置と行動を予測する手法について記述する。まず、線形補外について紹介し、それを用いて未来の対戦相手の位置を予測する方法を述べる。その後 k 近傍法について紹介し、どのように未来の対戦相手の行動を予測するのかを述べる。

4.1 線形補外

補外法 [6] は、離散点 $x_k (k = 1, 2, \dots, n)$ に対する関数 $y = f(x_k)$ が与えられたとき、区間 $[x_1, x_n]$ 外の点 x^* に対する y^* を求める数値計算の手法である。離散点が従う関数が線形の場合、この計算手法のことを特に線形補外と呼ぶ。離散点 $(x_1, y_1), (x_2, y_2)$ が与えられたとき (x_3, y_3) を求めたいとする。

これらの点が線形に変化すると分かっているとき、図 2 より以下の式を導き出せる。

$$\frac{x_3 - x_1}{x_2 - x_1} = \frac{y_3 - y_1}{y_2 - y_1} = (\text{定数})$$

未知数は x_3 と y_3 だけなので、どちらか一点でも分かればもう一点が分かる。もしくは $x_3 - x_1 : x_2 - x_1$ の比が分かっている場合はその比から x_3, y_3 を導き出すことが出来る。

4.2 k 近傍法

k 近傍法 [7] は、過去の数値データとそのときのクラスを保存しておき、新たな数値データが与えられるごとにそのデータから最も近い k 個のデータを選び多数決でクラスを選ぶ分類アルゴリズムである。ある数値データ x, y と、そのときのクラスの関係を記した図 3 を例として、分類方法を説明する。数値データの持つ値は 0 から 100 までの整数で、a または b のクラスを持っている。図中で青い点はクラスが a であることを、赤い点はクラスが b であることを表し、緑の点はクラスを求めたい新しい数値データ (30, 50) を表している。

新たな数値データ (30, 50) が与えられたとき、k 近傍法でこの数値データが持つクラスを求める。まず、全てのデータ点と新たなデータ点とのユークリッド距離を求める。図 3 から最も距離が近い点は (34, 48) の点であることが分かるので、k の値を 1 と選択すると、(34, 48) のクラスが b であることから (30, 50) のクラスは b と分類される。k = 2 の時は最も近い 2 点である (34, 48) と (29, 56) のクラスが b なので同様にクラスが b となるが、k = 4 の時点で近傍点のクラスは a と b の数が等しくなり、k = 5 の時、(30, 50) のクラスは多数決により a と決定される。

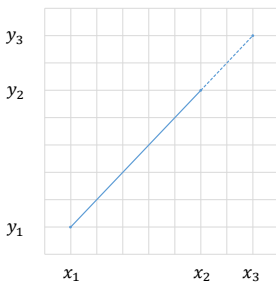


図 2: 線形補外

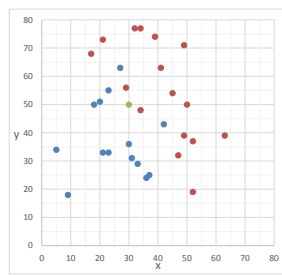


図 3: 数値データとクラス

4.3 提案手法

本研究では、線形補外を用いて対戦相手の位置を、k 近傍法を用いて対戦相手の行動を予測し、これらを組み合わせることで対戦相手の未来の情報を予測する手法を提案する。まず、対戦相手の現在と過去の位置座標から、未来の位置座標を補外法を用いて算出する。短時間の対戦相手の動きが線形であるという仮定をし、現在 (実際には 15 フレーム前) の位置 (x_0, y_0) と予測したいフレームの数 f だけ過去の位置 (x_f, y_f) との差分 (x_{diff}, y_{diff}) から未来の位置 (x_{pred}, y_{pred}) を計算する。 x_{pred} の計算式は以下ようになる。

$$x_{diff} = x_0 - x_f \quad (2)$$

$$x_{pred} = x_0 + x_{diff} \quad (3)$$

式 (2)、式 (3) より

$$x_{pred} = 2x_0 - x_f \quad (4)$$

が導かれる。 y_{pred} についても同様に、

$$y_{pred} = 2y_0 - y_f \quad (5)$$

が導かれる。

さらに対戦相手の行動を予測するために、 t フレーム目を基準として f フレーム過去の対戦相手との相対座標 (x_{t-f}, y_{t-f}) と $2f$ フレーム前との相対座標の変化量 $(x_{t-f} - x_{t-2f}, y_{t-f} - y_{t-2f})$ の 4 つの数値をデータとして、対戦相手の現在の行動 ($Action_t$) をクラスとして選び、1 フレームごとに対戦相手のデータの保存と行動の分類を行った。あるフレームで得られた現在 (t_c) の位置から f フレーム後の行動 ($Action_{t_c+f}$) を得ることが目的のため、対戦相手の f フレーム前の位置と現在の行動をデータセットとした。格闘ゲームにおいては対戦相手の絶対的な位置ではなく自分との間の距離が重要であることから、対戦相手の位置座標ではなく対戦相手との距離を数値データとして選択した。また f フレーム前の距離だけを数値データとした場合、現在の距離が全く違っていったときのクラスを同時に含んでしまうことが考えられる。このデータの重なりを防ぐために、 f フレーム前の距離だけでなく $2f$ フレーム前における距離の変化量も数値データとして選んだ。

5. 提案手法の検証

本章では、前章で提案した線形補外と k 近傍法による予測の有効性を検証する。まず予測フレーム数 f と対戦スコアの関係性を明らかにするため、線形補外による位置予測のみを適用したエージェントとコンペティション出場者である 6 種類のエージェントとの対戦について記述する。次に提案手法により NPC の対戦スコアが向上することを示すために、提案手法を適用したエージェントと上記の 6 種類のエージェントで行った対戦について述べる。

5.1 予測フレーム数 f の決定

格闘ゲーム FightingICE において、2014 年コンペティションの出場エージェントの一つである T3C に線形補外を用いた位置の予測を適用したエージェント (以降 coordinateT3C と呼ぶ) と同コンペティションの出場者である 6 体のエージェントを対戦させた。この際、 f の値を 0 から 29 まで 1 刻みで、30 から 100 まで 10 刻みで変化させ、各 10 試合、30 ラウンドずつ対戦を繰り返した。

図 4 は、予測フレーム数毎に 6 つの対戦スコアの平均を取り、適用前の T3C によるスコアとの差を取ったものである。元の T3C と比べてどれだけ多く対戦スコアを得られたかを表している。図 4 より、直近の予測と 20 フレーム前後の予測をしたときにより高い対戦スコアを得られることが分かった。この結果から、最もスコアが大きかった 2 フレーム (赤色で図示) と、20 フレーム付近の予測が有効に働く対戦相手を考慮してその付近で最もスコアが大きかった 16 フレーム (緑色で図示) を f のパラメーターとして決定した。

5.2 k 近傍法による行動の予測実験

前節により、位置の予測には $f = 2$ と $f = 16$ が適切という結果が得られた。本節では、さらに行動の予測をするため coordinateT3C に k 近傍法を適用したエージェント (kNNT3C と呼ぶ) とコンペティション出場者の 6 体のエージェントとの対戦を行った。k 近傍法のパラメーター k の値を 1 から 10 に変化させ、各 30 回ずつ対戦を繰り返した。

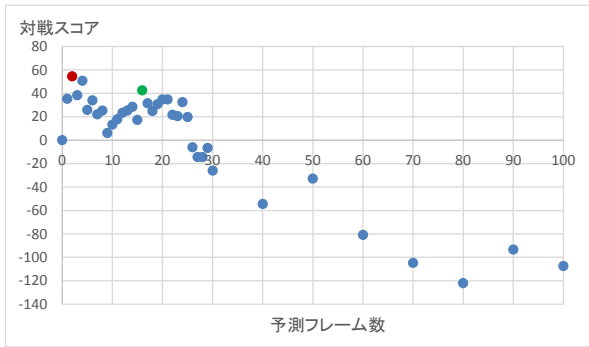


図 4: 各 f における T3C の対戦スコアと coordinateT3C の対戦スコアの差

5.3 実験結果

表 1 は各 f と各対戦相手 で coordinateT3C の平均スコアが最大となった k の値を表す。これらの k のとき、全てのラウンドでの対戦スコアの平均を表したグラフと有意差を示したものが図 5 である。提案手法を適用したエージェントは適用前と比べて 12 の組み合わせのうち、 $f = 16$ で Seal_Switch を相手にした時を除く 11 のパターンで対戦スコアを向上することが出来た。この差が有意であるかを検定したところ、有意水準を 5% とすると 16 のうち 7 つのパターンで、有意水準を 1% とすると 16 のうち 4 つのパターンでスコアの向上が有意と示された。

	T3C	ATTe	DKin	PasAI	S_Swi	SFig
$f = 2$	6	10	10	9	8	5
$f = 16$	1	9	9	7	1	10

表 1: 最も対戦結果が良かったときの k の値

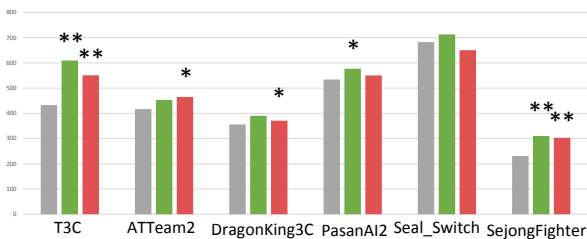


図 5: 提案手法適用前後の対戦スコアの比較。灰色、緑、赤の順に元の T3C、 $f = 2$ 、 $f = 16$ の時の対戦スコア。*、**はそれぞれ元の T3C と比べ有意水準 5% で差あり、有意水準 1% で差ありを示す。

6. 結論

本研究では、対戦型コンピュータゲームの一種である格闘ゲームを題材として、人間のプレイヤーが楽しめる強いエージェントを作る事を目標とした。まず、格闘ゲームが強いエージェントの鍵はスポーツと同じように状況判断能力であるという仮定をおき、それを検証すべく認識能力の遅れと対戦スコア

の関係性を調査した。結果として、例外はあるものの格闘ゲームにおいても認識能力の遅れが状況判断に悪影響を及ぼすということが分かった。

この結果を逆手に取り、対戦相手の未来の位置と行動を予測することで対戦スコアを改善することを目的として、線形補外法を用いて対戦相手の位置を、 k 近傍法を用いて対戦相手の行動を予測する手法を提案した。この手法を適用したエージェントが 6 種類のエージェントと戦って得た対戦スコアと、提案手法未適用のエージェントが同様のエージェントと戦って得た対戦スコアを比較したところ、2 フレーム後を予測した時はすべての対戦相手に対して、16 フレーム後を予測した時は 1 つを除いたすべての対戦相手に対して対戦スコアの向上が見られ、提案手法の有効性が確認できた。

今後の方向性として、本研究では情報の認識に主眼を置いてきたが、情報に対する行動選択を改善する、または行動の選択を学習するエージェントについて検討する余地がある。例えば、対戦相手が一定の距離に近づいた時に攻撃を放つという行動パターンを持つエージェントを考えると、対戦相手に攻撃が当たったかどうかを記録することで、攻撃が当たりやすい距離を学習することができる。強化学習を利用し、ある行動をしたときに相手に与えたダメージと自分の受けたダメージを報酬とすることで行動の選択を学習するといった手法も考えられる。また、コンペティションの出場エージェントの一つ Seal_Switch に対しては、16 フレーム後の予測がスコア向上を果たさず、2 フレーム後の予測もスコアの向上は有意ではなかった。この原因として、Seal_Switch は複数の行動規則を持ち必要に応じてそれを切り替えているため、予測が当てはまらなかったものと考えられる。このように、行動パターンが一つでないエージェントに対しても有効な手法が必要である。パターンの変化を検知することで別々に k 近傍法に用いるデータを記録しておき、現在のパターンに対応したデータを予測に用いることが解決策として考えられる。

参考文献

- [1] G. N. Yannakakis, and J. Togelius: A Panorama of Artificial and Computational Intelligence in Games, *IEEE Transactions on Computational Intelligence and AI in Games*, 2014. doi:10.1109/TCIAIG.2014.2339221
- [2] 遠藤 雅伸: デジタルゲームにおける“AI”の役割, *情報処理*, Vol.53, No.2, pp.146–152, 2012.
- [3] <http://www.ice.ci.ritsumei.ac.jp/~ftgaic/index.htm> (2015/01/29 参照)
- [4] <http://www.slideshare.net/ftgaic/2014-fighting-game-artificial-intelligence-competition> (2015/02/11 参照)
- [5] 石垣 尚男, 真下 一策, 遠藤 文夫: トップレベルのスポーツ選手の視覚機能と競技力の関係, *愛知工業大学研究報告*, Vol.27-A, pp.43–47, 1992.
- [6] 永坂 秀子: 計算機と数値解析, 朝倉書店, 1980.
- [7] 元田 浩, 津本 周作, 山口 高平, 沼尾 正行; データマイニングの基礎, オーム社, 2006.