

G-106

# 対戦型格闘ゲームにおける重み付け $k$ -substring tree 構造を用いた 相手の行動予測と予測精度の向上

## *Predicting the Opponent's Action Using the $k$ -Nearest Neighbor Algorithm and Substring Tree Structure in a Fighting Game*

中川 裕登<sup>†</sup>      山本 界人<sup>†</sup>      ターウオンマツ ラック<sup>‡</sup>      原田 智広<sup>‡</sup>  
Yuto Nakagawa      Kaito Yamamoto      Ruck Thawonmas      Tomohiro Harada

### 1. はじめに

多くの対戦型格闘ゲームにおいて、対戦相手の強さはゲームの楽しさや難易度に大きく影響している[1]。対戦型格闘ゲームでは、対人戦が主な遊戯目的だが AI が操作する NPC とも対戦することができる。しかし、個人のゲームに対する習熟性、行動決定などのユーザスキルが高いプレイヤーに対しては、より高度な AI が必要である。

本研究では、より高度な AI を実現するために、高いユーザスキルを持つプレイヤーのプレイスタイルに着目する。このようなプレイヤーは相手の過去の行動パターンや自身と相手との位置関係を考慮し、相手の次の行動を予測する。そして、予測した行動に適切な対処を行うことで対戦を有利に進める。高いユーザスキルを持つプレイヤーほど、これらの予測および対処能力が正確である。このようなプレイスタイルを AI 側で実現することにより高度な AI の実現を目指す。

本稿では、相手の次の行動を予測する部分に着目する。高度な AI を実現するために予測精度の向上を目的とし、 $k$  近傍法を用いた予測手法と substring tree 構造を用いた予測手法を組み合わせた weighted  $k$ -substring tree 構造を用いた予測手法を提案する。そして、Fighting Game AI Competition で採用されているプラットフォームを用いて他の AI と対戦する。対戦結果と予測精度を算出した結果から提案手法の評価を行う。

### 2. 関連研究

高度な AI を実現する方法として、AI が相手の行動を予測し対処を行う手法[2][3]がある。文献[2]では、相手の過去の行動パターンに着目し、substring tree 構造を用いて次の行動の予測を行う。文献[3]では、自身と相手との位置関係と距離関係に着目し、 $k$  近傍法を用いて予測を行う。他には、人間のプレイスタイルの模倣と学習を通して強い AI を作る手法[4]などがある。

### 3. ゲームルール

本章では実験対象である Fighting Game AI Competition の大会ルールについて述べる。Fighting Game AI Competition が採用しているルールでは 1 ラウンド 60 秒とし、計 3 ラウンドを 1 試合とする 2D 対戦型格闘ゲームである。このゲ

ームでは 1/60 秒を 1 フレームという単位で表し、1 フレーム単位で入力が処理される。そのため、AI は 1/60 秒という限られた時間で次の自身の行動を決定するための多くの計算を行わなければならない。また、このゲームの最大の特徴は AI に与えられるゲーム内の状況が最新ではないという点である。AI に与えられる情報は 15 フレーム前の情報であり、AI はその情報をもとに処理を行わなければならない。

### 4. 提案手法

予測精度を向上させるために、 $k$  近傍法を用いた予測手法と substring tree 構造を用いた予測手法を組み合わせた weighted  $k$ -substring 構造を用いた予測手法を提案する。 $k$  近傍法を用いた予測手法と substring tree 構造を用いた予測手法についてはそれぞれ 3.1 節および 3.2 節で説明する。

#### 4.1 $k$ 近傍法を用いた予測手法

予測を行うために、1 試合を通して相手が攻撃を行うたびに相手の攻撃の種類、その時点での自身と相手との相対座標と自身と自身の後ろ側のステージの端までの距離を取得する。そして、これらのパラメータを一つのデータとして 3 次元グラフ空間上に保存する。AI が予測を行う必要がある場合は、その時点での自身と相手との相対座標、自身と自身の後ろ側のステージの端までの距離を取得し、グラフ空間上にプロットする。そして、プロットしたデータから一定距離内に一定数以上のデータがプロットされていれば相手は攻撃行動を取ると判断する。相手が攻撃を行うと判断すると、 $k$  個の近傍データを参照する。参照したデータを一つのデータセット  $D_{k,m}$  として取得する。

#### 4.2 substring tree 構造を用いた予測手法

予測を行うために、1 試合を通して相手が攻撃を行うたびに相手の攻撃の種類を取得し保存し続け、文字列  $D_{seq}$  を更新する。AI 自身の攻撃が相手に当たった際に、その時点での  $D_{seq}$  を入力データ列として substring tree を作成する。作成前の substring tree は根ノードのみで子ノードを持たない。substring tree の作成には文献[5]の手法を用いる。substring tree の作成が終了したら、 $D_{seq}$  は破棄し初期化する。予測を行う際は、 $D_{seq}$  の中で新しいものから順に  $l$  個 ( $l=1,2,\dots,l$ ) のデータを用いる。 $l$  個のデータの中から  $m$  個 ( $m=l-1,\dots,1$ ) のデータを再帰的に用いて、作成された

<sup>†</sup> 立命館大学情報理工学研究科

Graduate School of Information Science & Engineering, Ritsumeikan University

<sup>‡</sup> 立命館大学情報理工学部

College of Information Science & Engineering, Ritsumeikan University

substring tree を探索し予測結果候補を  $D_{substring}$  として取得する。

例えば、ある時点での  $D_{seq} = "ABCD"$  であり、自身の攻撃が相手に当たった場合の作成後の substring tree を図 1 に示す。また、 $D_{seq} = "ABCD"$  であり  $l=3$  の場合の substring tree の探索の様子を図 2 に示す。この場合、順に "BCD", "CD", "D" と探索を行い、 $D_{substring} = \{A, A, A, B, B\}$  を得る。

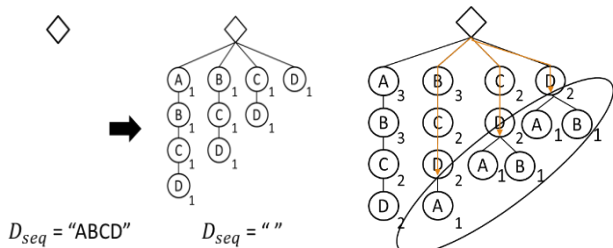


図 1 substring tree の作成

図 2 substring tree の探索

#### 4.3 weighted $k$ -substring tree 構造を用いた予測手法

$k$  近傍法、substring tree 構造を用いて得られたデータセット  $D_{knn}$ 、 $D_{substring}$  を用いて相手の次の攻撃予測を行う。まず、これらのデータセットを統合し新たなデータセット  $D_{k-substring}$  を取得する。また、 $D_{knn}$  のデータ数は一定であるのに対して、 $D_{substring}$  のデータ数は一定ではない。そのため、統合を行う際はデータセット内の各データに対して重みを付ける。重みはそれぞれのデータセットのデータ数の逆数で算出される。よって、 $D_{knn}$  の各データの重みは  $1/k$  となる。データセットの統合は攻撃の種類ごとに行い、各攻撃の種類ごとに重みの合計値を算出する。そして、重みが最大となる攻撃を相手が次に行う攻撃行動として予測を行う。

例えば、以下のような 2 つのデータセット  $D_{knn} = \{A, A, B\}$   $D_{substring} = \{A, B, C, C\}$  が得られた場合、各データの重みの値を算出し添え字で表すと、 $D_{knn} = \{A_{1/3}, A_{1/3}, B_{1/3}\}$ 、 $D_{substring} = \{A_{1/4}, B_{1/4}, B_{1/4}, C_{1/4}\}$  となる。よって統合後に得られるデータセットは  $D_{k-substring} = \{A_{11/12}, B_{10/12}, B_{1/4}\}$  となり、次の相手の攻撃は  $A$  であると予測する

## 5. 実験と結果

### 実験方法

提案手法を実装することにより、予測精度を向上させることができているかどうかを実験によって検証する。検証を行うために、 $k$  近傍法を用いて予測を行う AI、weighted  $k$ -substring tree 構造を用いて予測を行う AI を用意する。また今回の実験では、既存手法である、自身と自身の後ろ側のステージの端までの距離を用いずに 2 次元空間上でデータを扱う  $k$  近傍法 (2D-knn) を用いて予測を行う AI、substring tree 構造を用いて予測を行う AI を比較手法として用意した。それに加えて、重み付けを行わずに  $D_{k-substring}$  内のデータから多数決で予測を行う  $k$ -substring tree 構造を用いて予測を行う AI を用意した。 $k$  近傍法と substring tree 構造の組み合わせは、上記の 2 種類の  $k$  近傍法に対して行った。また、 $k$  近傍法で用いるパラメータ  $k$  の値は 3 に固定した。substring tree 構造で用いるパラメータ  $l$  の値は 3 に固定して実験を行った。AI は対戦中に相手の次の攻撃を予測する。予測精度は以下の式(1)によって求められる。

$$PredictionAccuracy = \frac{successPredictionNumber}{totalOppActionNumber} * 100 [\%] \quad (1)$$

$successPredictionNumber$  は予測が成功した相手の攻撃数であり、 $totalOppActionNum$  は相手が 1 試合中に行った全ての攻撃の合計数である。

実験は上記の AI 計 7 体と Fighting Game AI Competition の 2014 年度、3C 部門に参加した 8 体の AI との対戦を行う。対戦は各 AI に対して 10 試合行い、予測精度の平均値を比較する。なお、予測を行ったフレームから  $n$  フレーム以内の次の相手の攻撃が予測結果と一致していれば予測成功とする。

### 実験結果

表 1 に 8 体の AI と対戦した際の予測精度の平均値を示す。また  $n=3600$  はそのラウンド内での次の相手の攻撃を予測結果と一致しているかを確認する。

表 1 全 AI に対する予測精度の平均値 (10 試合)

$n$	0	15	30	3600
2D-knn	23	45	52	57
knn	21	46	54	58
substr	11	15	18	25
2D-k-substr	21	45	51	55
k-substr	22	46	53	57
weighted 2D-k-substr	22	46	54	59
weighted k-substr	24	48	56	60

表 1 より、提案手法を実装することにより予測精度が向上していることが分かる。substring tree 構造のみを用いた予測手法は予測精度が比較的低いことがわかる。

## 6. おわりに

本稿では、予測精度を向上させる手法と AI の性能を向上させる手法を提案した。 $k$  近傍法を用いた予測手法と substring tree 構造を用いた予測手法を組み合わせることにより、予測精度を向上させることができた。substring tree 構造を用いた予測手法の予測精度が高くなかった原因としては、今回の実験で使用したプラットフォームである FightingICE には明確に定義された一連の攻撃行動であるコンボが存在しなかったためであると考えられる。今回の実験では、予測結果と相手の次の攻撃行動が一致しているかどうかで予測精度を算出した。今後は、似たような攻撃行動をグループに分類し、グループ単位で予測結果が一致しているかどうかで予測精度を算出する。

### 参考文献

- [1] Jenova Chen: "Flow in Game," Communications of the ACM, Vol. 50, No. 4, pp. 31-34, 2007.
- [2] Yuto Nakagawa, Kaito Yamamoto, and Ruck Thawonmas, "Online Adjustment of the AI's Strength in a Fighting Game Using the k-Nearest Neighbor Algorithm and a Game Simulator," Proc. of the 3rd IEEE Global Conference on Consumer Electronics (GCCE 2014), Tokyo, Japan, Oct. 7-10, 2014.
- [3] Moriyama, Koichi, et al. "An Intelligent Fighting Videogame Opponent Adapting to Behavior Patterns of the User." IEICE TRANSACTIONS on Information and Systems 97.4 (2014): 842-851.
- [4] S, Lueangruangroj and V. Kotrajaras, "Real-time Imitation based Learning for Commercial Fighting Games," Proc. of Computer Games, Multimedia and Allied Technology 2009, pp. 1-3, 2009.
- [5] Cao, Huiping, Nikos Mamoulis, and David W. Cheung. "Mining frequent spatio-temporal sequential patterns." Data Mining, Fifth IEEE International Conference on. IEEE, 2005.