

# デジタルカーリングにおける局面に対する 期待得点分布の学習

安宅 耕太郎<sup>1,a)</sup> 飯塚 博幸<sup>2</sup> 山本 雅人<sup>2</sup>

概要：デジタルカーリングはカーリングの戦略をコンピュータ上で解析するために開発されたシミュレータである。低確率で高得点が狙える場合と高確率で低得点を狙える局面が区別できないため、獲得点数の期待値だけでは正確な局面価値の評価には不十分である。局面遷移に不確定性をもつデジタルカーリングにおいて、手堅く点数を獲得できるかどうかは最終的な勝敗に深く関わっているため、期待得点分布を正しく推定できることが重要である。本論文では、任意の局面に対する期待得点分布をニューラルネットワークにより学習を行う手法について提案する。従来手法と比べて、ニューラルネットワークの入力表現に各ストーンのハウスの中心からの距離を加えることにより、既存モデルよりも正確な予測が可能となることを示す。

## 1. はじめに

カーリングは、氷の上で2つのチームが交互にストーンを投げ、最終的な総得点を競うスポーツであり、冬季オリンピックの正式種目に指定されている。各チーム8回ずつの投球することをエンドと呼び、エンドを繰り返すことで試合が進行する。各エンドの終了時には、得点計算が行われる。カーリングでは、正確にストーンを投げるという技術的要素だけでなく、得点差、残りのストーン数、ストーン配置などの様々な要因を考慮した上でどこへ投球するかという戦略的要素が重要である [1]。そこで、電気通信大学の伊藤らのグループによってカーリングをコンピュータ上で解析するためにデジタルカーリングが開発された [2][3]。デジタルカーリングにおいては、二人のプレイヤーが交互にストーンの初速度と回転方向を入力することで、ストーンが投げられゲームが進行する。ストーンの初速度と盤面のストーンの座標は連続値であるため、次の一手として候補となる初速度や局面は理論上無限に存在する。さらに、現実のカーリングの不確定性を再現するためにプレイヤーの選んだ初速度に乱数を加えている。

加藤ら [4] は、囲碁や将棋などの一般的なボードゲームに用いられているゲーム木探索という手法を、連続かつ不確

定性をもつデジタルカーリングに適用し、勝率を上げることに成功している。さらに、Yamamoto, et al.[5] はゲーム木探索に用いる局面評価関数の精度を向上させるために、カーリングにおいては最終的な総得点が1点でも多ければ勝ちであり、得点期待値の高い局面が良い局面であるとは限らないという点に着目した。例えば、同点で最終エンドである場合、「高得点を得られる可能性が高いが失点の可能性も高い局面」よりも「1点しか得られそうにないが失点する可能性がほぼない局面」の方が価値が高いと判断できる。このように局面の価値を正しく評価するためには、エンド終了時に得られる得点の確率分布（期待得点分布）を予測し考慮することが重要である。Yamamoto, et al.[5] は期待得点分布を予測するためにニューラルネットワークを用いて学習を行っていたが、局面の入力表現が不十分であり改善の余地がある。

本研究では、既存モデルの局面の入力表現に、新たに各ストーンのハウスの中心からの距離を加えることで、より高精度な期待得点分布を推定可能とする手法を提案する。

## 2. カーリング

### 2.1 カーリングのルール

カーリングは、2チームが交互にストーンを投げ、最終的な総得点を競うスポーツである。エンドを繰り返すことで試合が進行し、各エンドの終了時には得点計算が行われる。

相手チームの、ハウスの中心に最も近いストーンよりも内側に存在する自分のストーンの個数が得点として与えられる。ただし、ハウス外にあるストーンは得点計算におい

<sup>1</sup> 北海道大学大学院 情報科学院  
Graduate School of Information Science and Technology,  
Hokkaido University

<sup>2</sup> 北海道大学大学院 情報科学研究院  
Faculty of Information Science and Technology,  
Hokkaido University

a) ataka@complex.ist.hokudai.ac.jp

て考慮しない。また、ハウス内にストーンがひとつもない場合は各チーム0点とする。

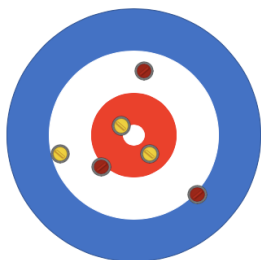


図1 得点計算の例（黄チーム2点，赤チーム0点）

## 2.2 デジタルカーリング

デジタルカーリングは、コンピュータ上でカーリングをプレイすることを可能としたシミュレータである。ルールや試合の進行は、現実のカーリングに沿って行うが、ストーンの投球はコンピュータ上のシミュレーションによって行う。デジタルカーリングにおける投球のシミュレーションは、物理シミュレータと乱数生成器によって実現されている。

物理シミュレータは、式(1)に示すように、ストーンの初速度  $v$ 、回転方向  $\theta$ 、と現在の局面  $S_t$  を入力とし、投球後のストーン軌跡および衝突などの物理演算を行い、投球した後の局面  $S_{t+1}$  を生成する。

$$S_{t+1} = f(v, \theta, S_t) \quad (1)$$

$v$  は2次元直交座標系の実数値ベクトルである。回転の速さは一定値としているので、回転方向  $\theta$  は右回転か左回転のいずれかである。 $S_t$  は現局面までに投げられたストーンの数、現局面の手番、現局面のエンド数、全エンド数、各エンドの得点、各ストーンの座標を情報としてもつ。衝突が起こらなかった場合にストーンが到達する座標  $p'$  とすると、すべての  $\theta$  に対して  $v$  と  $p'$  は1対1に対応しており、以下のように変換可能である。

$$(v, \theta) = g(p', \theta) \quad (2)$$

$$(p', \theta) = g^{-1}(v, \theta) \quad (3)$$

現実のカーリングにおける手ブレや氷の表面状態などの不確定性を再現するために、乱数生成器によって二次元ベクトル  $v_{rnd}$  を生成し、プレイヤーが入力するストーンの初速度  $v_{in}$  へ加える。これにより、同じ  $v_{in}$  から異なる結果が生じうる。

$v_{rnd}$  の生成は、以下の手順のように到達目標地点に誤差を加え、それに対応する速度ベクトルの差を求める方法が、直近のデジタルカーリングの大会で採用されている。しかし、この方式は現実のカーリングにおいて投げる方向

と投げる強さに対し生じる誤差の表現として不自然であるという指摘を受け、極座標表示した速度ベクトルに対し正規分布に従う誤差を加える方式も提案されている [6]。

- (1) 平均0、標準偏差  $\sigma_x, \sigma_y$  の正規分布に従う確率変数  $r_x, r_y$  を要素に持つ二次元ベクトル  $r$  を生成。
- (2) ハウス中央の座標  $p_h$  を投球目標座標とする初速度  $v_h$  を算出。

$$(v_h, \theta) = g^{-1}(p_h, \theta) \quad (4)$$

- (3)  $p_h$  から  $r$  ずれた点を投球目標座標とする初速度  $v'_h$  を算出。

$$(v'_h, \theta) = g^{-1}((p_h + r), \theta) \quad (5)$$

- (4)  $v'_h$  と  $v_h$  の差分ベクトルを誤差ベクトル  $v_{rnd}$  とする

$$v_{rnd} = v'_h - v_h \quad (6)$$

デジタルカーリングにおいて投球シミュレーションの流れは、プレイヤーがデジタルカーリングにストーンの初速度  $v_{in}$  と回転方向  $\theta$  を入力した後、乱数生成器により生成した  $v_{rnd}$  を  $v_{in}$  へ加え、物理シミュレータへ入力することで式(7)により次の局面  $S_{t+1}$  を得る。

$$S_{t+1} = f((v_{in} + v_{rnd}), \theta, S_t) \quad (7)$$

$v_{in}$  がハウス中央を投球目標座標とする初速度である場合、 $g(v_{in}, \theta)$  により得られる投球目標座標  $p_{in}$  に対する  $g((v_{in} + v_{rnd}), \theta)$  により得られる投球目標座標  $p'_{in}$  のずれ  $p'_{in} - p_{in}$  は  $r$  に従う。 $v_m$  がハウス中央以外を投球目標座標とする初速度である場合も同様となる。 $r$  の各要素  $r_x, r_y$  が従う正規分布の標準偏差  $\sigma_x, \sigma_y$  は、公式大会においては予め公開されているため既知の情報として利用できる。

## 3. 期待得点分布の学習

期待得点分布とは、エンド終わりに得られる得点の確率分布である。不確定ゲームにおいて、現在の局面価値は、1手先の局面の評価値とその局面への遷移確率を掛けて足し合わせた期待値によって表される。図2のように、期待値が大きい局面であっても、相手が得点を得る可能性がある局面が存在する。特に同点であったり、自分のチームがリードしている場合は、相手が得点を得る可能性のある選択肢は選ばないようにすべきである。期待得点分布を評価関数に組み込むことによって、このようなりスクを判別できるようになる。具体的には、最善手の選択には勝率テーブルを用いる。勝率テーブルとは、残りエンド数と得点差が与えられたときに期待勝率を表すテーブルである。期待勝率はAI同士の自己対戦の結果から予め計算される。期待得点分布と勝率テーブルの値を参照することで、期待勝率が得られるため最善手の選択を行うことができる。

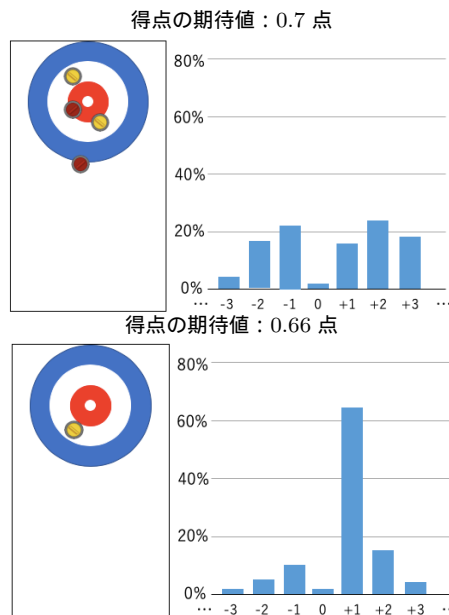


図 2 期待得点分布の例．黄チームが先手，赤チームが後手とし，赤チームの得点を正とする．

本論文では，カーリング AI 同士の自己対戦から 15 投を終えた局面を抽出し，その局面に対する期待得点分布を実際にその局面から多くのシミュレーションをすることによって算出し，生成した局面と期待得点分布の組を用いてニューラルネットワークを学習させる．

既存モデルでは入力にハウスの中心を原点とする座標系で石の座標を含んでいるが，石がプレイエリアに存在しない場合には 0 を入力していた．ハウスの中心に近い状態と，プレイエリア外にある状態が入力として似てしまうため適切でないと考えられる．本論文では，プレイエリア外にある石の座標を 0 でなく具体的な値を入力するなどの工夫を行った．

得点差\残りエンド	4	3	2	1	0
3	88.7%	91.9%	94.6%	96.2%	100.0%
2	70.7%	77.1%	79.4%	88.1%	100.0%
1	54.5%	60.9%	55.7%	67.7%	100.0%
0	34.2%	34.0%	27.9%	26.0%	22.0%
-1	18.9%	16.2%	12.2%	4.2%	0.0%
-2	6.5%	3.4%	2.1%	1.1%	0.0%
-3	1.5%	0.0%	1.4%	1.1%	0.0%

図 3 勝率テーブルの例

### 3.1 ニューラルネットワークによる期待得点分布の推定

図 4 のように，既存モデルと同様に局面情報を入力として，期待得点分布を出力するニューラルネットワークを用いる．ニューラルネットワークの出力は，それぞれの得点に対しての確率である．最大 8 点まで獲得する可能性があり，どちらの石もハウス内に存在せず 0 点になる可能性もあるため，相手が得点する場合を負の値で表すとし出力は -8 以上 8 以下の 17 通りの整数値をとる．

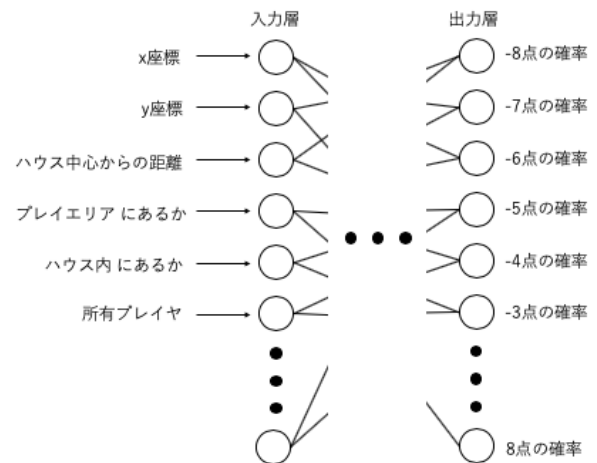


図 4 ニューラルネットワークの入力層と出力層

#### 3.1.1 入力する局面情報

入力とする局面は 15 投目を投げ終えた状態である．表 1 に既存のモデルの入力を示す．提案モデルの入力は，表 2 に示すように投げ終えた 15 個のストーンそれぞれに対して， $x$  座標， $y$  座標，所有者，ハウスの中心との距離，プレイエリアにあるか，ハウス内にあるかという 6 種類の数値を，ハウスの中心に近い順番に並び替えた合計 90 次元を入力とする．

$x$ 座標	[-2.23, 2.23]
$y$ 座標	[-1.975, 6.4]
プレイエリアにあるか	0, 1
ハウス内にあるか	0, 1
所有プレイヤー	0, 1, 2

表 1 既存モデルのストーン 1 個分の入力情報

$x$ 座標	[-2.23, 2.23]
$y$ 座標	[-1.975, 6.4]
ハウス中心からの距離	[0, 6.777]
プレイエリアにあるか	0, 1
ハウス内にあるか	0, 1
所有プレイヤー	-1, 0, 1

表 2 提案モデルのストーン 1 個分の入力情報

### 3.2 教師データの生成

#### 3.2.1 局面生成

実戦で現れやすい局面を生成するために，入力とする局面の生成はカーリング AI の自己対戦によって行う．デジタルカーリングの大会である UEC 杯で優勝経験のある「歩」というプレイヤーを用いた [7][8]．自己対戦の記録の中から，15 投目を投げ終えた局面のみを抽出し，合計 87,000 個の局面を作成した．一部の局面については，先攻の最初の投球をセンターガードにあえて置いた局面を生成したのち，「歩」によって自己対戦させて 15 投目まで投げ終えた

局面も生成した。これは、「歩」のみの自己対戦の場合、先攻側がセンターガードを置くような試合展開になりにくい性質を考慮し、より多くの試合展開になるような局面を学習する目的のために行った。

### 3.2.2 期待得点分布の生成

生成した局面に対する候補手のうち、外乱なしでストーンを投げたときに得点が最大となるようなショットを求め、候補手は、リンクのプレイエリアよりも少し奥に広い範囲を  $x$  軸方向に 40,  $y$  軸方向に 60 個に分割したときに、それぞれのグリッドの中心にストーンが届くようなショットとする。左回転か右回転かの 2 通りあるため、候補手は合計 4,800 通り存在する。得点が最大となるショットの初速度に乱数を加えて行うシミュレーションを繰り返すことによって、教師データとしての期待得点分布を生成する。生成した局面 1 対 1 に対応するので、同数の合計 87,000 個の期待得点分布が生成される。

### 3.3 学習方法

全体で 87,000 個の教師データのうち 9 割の 77,333 個を訓練データ、残り 1 割の 9,667 個をテストデータとする。バッチ数 32 のミニバッチ学習をおこなう。表 3 に示すように、入力の次元は 90, 出力の次元は 17 である。中間層は 3 層、ノード数は 50 個、中間層の活性化関数は ELU[9] を用い、出力層の活性化関数には softmax 関数を用いた。

入力層	90
中間層 1	50
中間層 2	50
中間層 3	50
出力層	17

表 3 ニューラルネットワークの各層のニューロン数

### 3.4 学習結果

#### 3.4.1 損失関数

生成したデータを用いる学習を 1500 エポックまで行った結果を図 5 に示す。縦軸はテストデータに対する二乗誤差関数の値、横軸はエポック数である。

損失関数の値がより小さくなっているため、提案モデルの方が教師データに近い期待得点分布を出力するということがわかる。

ただし、どちらの手法においても、数百エポックくらいから損失関数の値が減少せず、停滞、もしくは、悪化してのが見られ、過学習が起きていると考えられる。

#### 3.4.2 出力の比較

上記のように学習した既存モデルと提案モデルのうち損失関数の値が最小であるエポックのモデル（既存モデルは 100 エポック目、提案モデルは 600 エポック目）を用いて出力する期待得点分布と、学習時に教師データとした期待

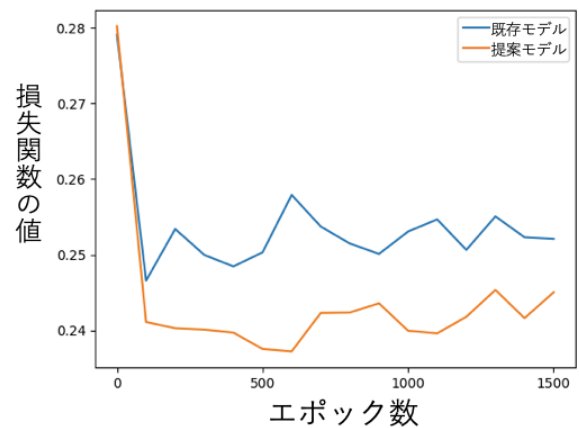


図 5 学習結果

得点分布との比較を行う。

図 6 に例として用いた局面を示す。この局面は、赤チームが最後のストーンを投球する。右上の赤チームのストーンはバックラインをまたがっているが、ストーンの一部がラインの内側に残っているためプレイエリア内にあるとみなされる。図 7 からわかるように、既存モデルの出力は 4 点以上の点数をとる確率を大きく見積もっている。このことから、既存モデルは距離の値を明示的に与えていないので、「 $y$  座標が  $x$  座標がハウスに近いとそのストーンも得点に結びつく」と誤って認識している可能性が高いことがわかる。

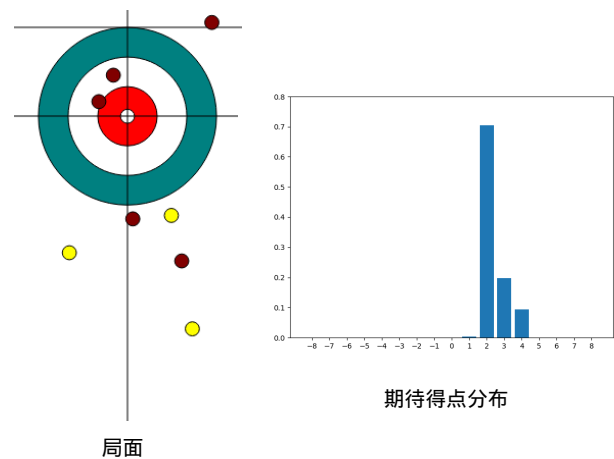


図 6 正解データの局面と期待得点分布

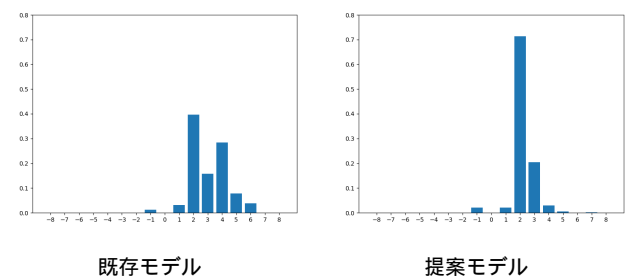


図 7 既存モデルと提案モデルの出力の期待得点分布

### 3.4.3 考察

提案モデルは既存手法の中間層の数、活性化関数などが同一であるにもかかわらず、損失関数の値が既存モデルよりも小さくなっていることから、提案モデルは、ニューラルネットワークの入力にそれぞれのストーンハウスの中心からの距離を明示的に加えることで、局面に対する期待得点分布をより正確に評価可能になったといえる。また、それぞれのモデルの出力を比較することで、距離を入力表現に加えることが、ハウス外のストーンが追加得点にあまり貢献しないことを学習するのに役に立つことがわかった。

## 4. おわりに

本論文では、既存の期待得点分布推定モデルの入力表現に各ストーンハウスの中心からの距離を追加し、提案モデルに対して、新たに生成した局面と期待得点分布の組を用いて学習した。損失関数の値によって性能を比較したところ、提案モデルの方が損失関数の値が小さくなっていることから、既存モデルよりも正確な期待得点分布を学習できていることが示された。

## 参考文献

- [1] Coleman, G.: Introduction to Curling Strategy. (2014).
- [2] 北清勇磨, 岡田雷太, 伊藤毅志: デジタルカーリングサーバーの提案と紹介, 情報処理学会研究報告, 2014-GI-31, No. 2, pp. 1 - 5 (2014).
- [3] 電気通信大学 . デジタルカーリング, <http://minerva.cs.uec.ac.jp/curling/wiki.cgi> (2019. 6. 22 閲覧).
- [4] 加藤修, 飯塚博幸, 山本雅人: 不確定性を含むカーリングにおけるゲーム木探索, 情報処理学会論文誌, Vol. 57, No. 11, pp. 2354-2364 (2016)
- [5] Yamamoto, M., Kato, S., Iizuka, H.: Learning of Expected Scores Distribution for Positions of Digital Curling. Workshop on Curling Informatics. (2018).
- [6] 森健太郎, 伊藤毅志, 条件変更にロバストなデジタルカーリングの改良, 情報処理学会ゲーム情報学研究会報告, GI-41(11)(2019).
- [7] 大渡勝己, et al. カーリング AI に対するモンテカルロ木探索の適用. ゲームプログラミングワークショップ 2016 論文集, 2016, 2016: 180-187.
- [8] 伊藤毅志, 森健太郎: デジタルカーリング大会報告 2015 年度, 情報処理学会研究報告, 2016-GI-36, No. 2, pp. 1 - 6 (2016).
- [9] Clevert, Djork-Arn, Thomas Unterthiner, and Sepp Hochreiter. "Fast and accurate deep network learning by exponential linear units (elus)." arXiv preprint arXiv:1511.07289 (2015).