Vol.2014-GI-31 No.15
2014/3/17

大貧民における棋譜データからの提出手役評価関数の学習

地曳 隆将[1,a)]　松崎 公紀[2,b)]

概要：棋譜を教師データとした評価関数の学習は，特にコンピュータ将棋において有効とされている．本研究では，コンピュータ大貧民を対象として，棋譜を教師データとした提出手役評価関数の学習を行いその性能を評価した．提出手役評価関数には 3 層ニューラルネットワークを用いた．提出手役評価関数の性能を評価するため，棋譜で提出した手役との一致率を調査した．その結果，学習に使用する教師データを増やすことで一致率が上昇したが，教師データ数 15000 程度で一致率が頭打ちになることが確認された．教師データ数 15000 の棋譜評価関数では，未知の盤面に対する提出手役一致率が 69%であった．

キーワード：大貧民，3 層ニューラルネットワーク，評価関数，モンテカルロ法

1. はじめに

本研究では，多人数不完全情報ゲームである大貧民を研究の対象とし，棋譜によって学習をした提出手役評価関数の性能を調査する．さらに，提出手役評価関数をモンテカルロ法プレイヤに適用することでプレイヤの強化を図る．これまでに大貧民では，モンテカルロ法を適用したプレイヤの有効性が示されており，実際に UEC コンピュータ大貧民大会（UECda）[13] で優勝するレベルのプレイヤが作成されている [8], [9], [10].

大貧民においてモンテカルロ法プレイヤを強化するためには，プレイアウト回数を増やすだけでなく，プレイアウトの精度を高める必要がある．プレイアウトの精度を下げる要因のひとつとして，プレイアウト中の提出手役の差がある．原始モンテカルロ法プレイヤのプレイアウトでは，各プレイヤの提出手役の選択を乱数を用いて等確率に行う．そのため，実際のゲームにおいて選択しないであろう手役を提出した場合もシミュレーションしてしまうことがあるため，プレイアウトの精度が低下する要因となる．

そこで本研究では，プレイアウト中の各プレイヤの提出手役の選択を評価関数を用いて行うようにする．これにより，手役選択が実際のゲームに近づきプレイアウトの精度が向上できると考える．提出手役評価関数には 3 層ニューラルネットワークを用い，その重みを棋譜によって調整する．そして，この提出手役評価関数をモンテカルロ法プレイヤのプレイアウト部分に適用することで，プレイアウト精度の向上を図る．

本研究では提出手役評価関数を 2 つの観点で性能調査する．まず，提出手役評価関数自体の性能調査として，棋譜で提出された手役と，提出手役評価関数が示す最善手がどの程度一致するかを調査する．次に，提出手役評価関数を適用したモンテカルロ法プレイヤの性能調査として，対戦を行い強さを評価する．

本論文の貢献は，大きく次の 3 点である．

- 3 層ニューラルネットワークを用いた提出手役評価関数の性能について調査した．
- 大貧民に対する 3 層ニューラルネットワークの適用を設計した．
- 提出手役評価関数を適用したモンテカルロ法プレイヤの強さを調査した．

本論文の構成を以下に示す．第 2 章では，関連研究について述べる．第 3 章では，モンテカルロ法プレイヤについて説明する．第 4 章では，3 層ニューラルネットワークについて説明する．第 5 章では，3 層ニューラルネットワークを用いた提出手役評価関数の性能を調査する．第 6 章では，提出手役評価関数を適用したモンテカルロ法プレイヤの性能を調査する．そして，第 7 章で本論文をまとめる．

2. 関連研究

大貧民ではプレイヤ強化の一環として，プレイアウト中の提出手役を実際のプレイヤに近づける研究も行われてい

[1] 高知工科大学大学院工学研究科基盤工学専攻
Graduate School of Engineering, Kochi University of Technology
[2] 高知工科大学情報学群
School of Information, Kochi University of Technology
[a)] 165064w@gs.kochi-tech.ac.jp
[b)] matsuzaki.kiminori@kochi-tech.ac.jp

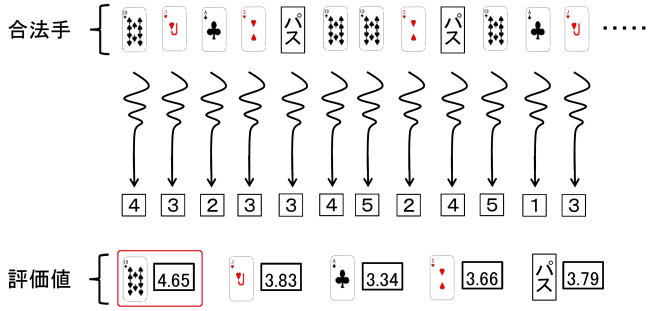

図 1 モンテカルロ法プレイヤの動作

る．伊藤ら [3] は，実際のプレイヤの提出手役を模倣する手法としてナイーブベイズを用い，過去の UECda 優勝クライアントである snowl に対して提出手役一致率がおよそ 4 割であったと報告している．

棋譜を利用して着手選択に用いる評価関数の学習を行う研究も行われている．1950 年代にオセロにおいて，ゲームの棋譜を利用して線形和で表現された評価関数の重みを調整する方法が提案された [1]．棋譜による評価関数の学習は主に将棋で有効性を示している．2006 年に保木 [11] が発表した手法では，評価関数を棋譜で指された手を出すように重みを調整することで 10,000 を超える重みの調整に成功し，コンピュータ将棋選手権で優勝するレベルのプレイヤが得られた．評価関数の調整に棋譜を用いる手法は近年の将棋プレイヤの開発では広く用いられており [4], [5], [6]，2013 年に行われた第 2 回将棋電王戦においては，プロ棋士に勝利するレベルのプレイヤが複数作成されている [12]．

3 層ニューラルネットワークを用いてプレイアウト中の相手着手を模倣する研究は，大貧民と同じ多人数不完全情報ゲームの麻雀でも行われている．北川ら [7] は，3 層ニューラルネットワークを用いた評価関数を棋譜（牌譜）を用いて学習を行った．結果として評価関数を用いたプレイヤのレーティングは 1318 と弱かったが，棋譜と評価関数の打牌・行動一致率はツモ局面でおよそ 56%となり，鳴き局面においておよそ 89%となったと報告している．

3. モンテカルロ法プレイヤ

大貧民におけるモンテカルロ法プレイヤの動作を図 1 に示す．まず，モンテカルロ法プレイヤは，プレイアウトをする仮想的な盤面を生成する．大貧民では相手手札を知ることができないため，相手手札を残存カードから割り当てる必要がある．本実験で用いるモンテカルロ法プレイヤでは，相手手札は残存カードからランダムに割り当てた．

次に，モンテカルロ法プレイヤは，プレイアウトによって合法手の評価値を算出する．合法手に対して指定回数のプレイアウトを行い，最終的に最も評価値の高い手役を場に提出する．1 回のプレイアウト終了時に割り当てる点数は UECda の標準ルールにならって，大富豪であれば 5 点，富豪であれば 4 点，平民であれば 3 点，貧民であれば 2 点，

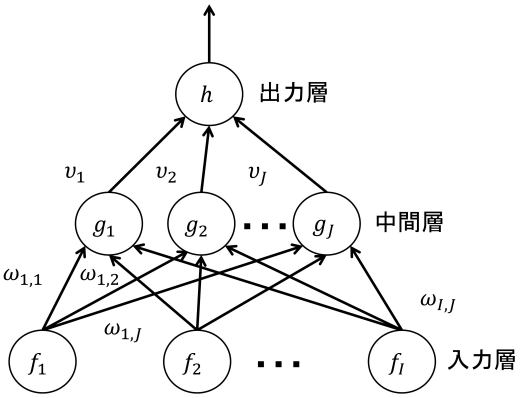

図 2 3 層ニューラルネットワーク

大貧民であれば 1 点とする．

プレイアウトの対象となる合法手には（全く意味のない，場が新しい場合を除き）パスも含める．プレイアウトでは，各プレイヤは以下のように動作する．

(1) 合法手のうち，それを出すことであがりとなるような手があれば，それを選択する．
(2) パス以外の合法手が存在する場合には，パス以外の合法手の中から等確率に選択する．
(3) 合法手がパスのみである場合には，パスを選択する．

上記のとおり，プレイアウトの対象となる最初の 1 手のみパスを含み，プレイアウト中では意図的なパスはしないとする．

プレイアウトする手役は UCB1 値 [2] が最大のものを選択する．UCB1 値は手役 j の評価値を $\overline{X_j}$，全体のプレイアウト回数を n，手役 j に対するプレイアウト回数を n_j，バランスパラメータを c として

$$\overline{X_j} + c\sqrt{\frac{2\log n}{n_j}}$$

で表される．UCB1 値のバランスパラメータ c は 1 回のプレイアウトで得られる報酬の最大値と最小値の差，すなわち 4 とした．

4. 3 層ニューラルネットワーク

図 2 に 3 層ニューラルネットワークの構成を示す．3 層ニューラルネットワークは，入力層，中間層，出力層の 3 層からなる．入力層に値を与えることで中間層を経て出力層から値が得られる．

入力層のユニット i から中間層のユニット j への遷移に対する重みを $\omega_{i,j}$，中間層のユニット j から出力層のユニットへの遷移に対する重みを v_j とする．$\omega_{i,j}$ の総数は，入力層のユニット数 I に中間層のユニット数 J を掛けた数に等しく，v_j の総数は中間層のユニット数 J に等しい．$\omega_{i,j}(1 \leq i \leq I, 1 \leq j \leq J)$ をまとめて ω と表す．同様に $v_j(1 \leq j \leq J)$ をまとめて v と表す．関数 $S(x)$ をシグモイド関数

$$S(x) = \frac{1}{1+\mathrm{e}^{-x}}$$

とする．また，関数 $D(x)$ を

$$D(x) = x(1-x)$$

とする．ここで，$S'(x) = D(S(x))$ が成り立つ．

各層の計算について以下に示す．入力層の各ユニットでは入力 p によって値が計算される．入力層のユニット i で計算される値を $f_i(p)$ とする．中間層の各ユニットでは入力層の値 $f_i(p)$ と $\omega_{i,j}$ によって値が計算される．中間層のユニット j で計算される値 $g_j(p,\omega)$ は

$$g_j(p,\omega) = S\left(\sum_{i=1}^{I} f_i(p)\omega_{i,j}\right)$$

である．出力層では中間層の値 $g_j(p,\omega)$ と υ_j によって値が計算される．3 層ニューラルネットワークの出力値 $h(p,\upsilon,\omega)$ は

$$h(p,\upsilon,\omega) = S\left(\sum_{j=1}^{J} g_j(p,\omega)\upsilon_j\right)$$

となる．

各層の重みの調整は，多数の教師データを用いて誤差逆伝播法によって行う．教師データは正解の入力 q_m と不正解の入力 p_m の組であり，教師データの数を M とする．教師データの集合を P と表す．ここで，正解の出力と不正解の出力の差の度合いを示す関数 $E(P,\upsilon,\omega)$ を

$$E(P,\upsilon,\omega) = \sum_{m=1}^{M} S\left(h(p_m,\upsilon,\omega) - h(q_m,\upsilon,\omega)\right)$$

と定める．誤差逆伝播法による重み調整では，$E(P,\upsilon,\omega)$ を最小化することを目標とする．

誤差逆伝播法による重み調整を以下に示す．あるときの調整前の重みを υ，ω，調整後の重みを υ'，ω' とする．学習率 η を用いて，υ，ω の重みを

$$(\upsilon',\omega') = (\upsilon,\omega) - \eta\frac{\partial E(P,\upsilon,\omega)}{\partial(\upsilon,\omega)}$$

によって調整する．ここで，υ_j の重みを調整するための勾配は

$$\alpha_{i,j}(p,\upsilon,\omega) = D(h(p,\upsilon,\omega))g_j(p,\omega)$$

を用いて

$$\frac{\partial E(P,\upsilon,\omega)}{\partial \upsilon_j} = D(E(P,\upsilon,\omega)) \cdot \sum_{m=1}^{M}\left(\alpha_{i,j}(p_m,\upsilon,\omega) - \alpha_{i,j}(q_m,\upsilon,\omega)\right)$$

によって計算される．$\omega_{i,j}$ の重みを調整するための勾配は

$$\beta_{i,j}(p,\upsilon,\omega) = D(h(p,\upsilon,\omega))\upsilon_j D(g_j(p,\omega))f_i(p)$$

```
Order Normal
Lock_Suits H
Last_Meld Single:H5
Can_Play [Yes,Yes,Yes,No,No]
Cards_Num [9,11,6,6,5]
Remaining_Cards S3 S4 H4 S5 H6 S6 C7 D8 S8 …
Player0_Cards C3 S7 H7 D7 H8 C8 SJ HQ DQ
Put_Meld Single:H8
```

図 3 盤面データ

を用いて

$$\frac{\partial E(P,\upsilon,\omega)}{\partial \omega_{i,j}} = D(E(P,\upsilon,\omega)) \cdot \sum_{m=1}^{M}\left(\beta_{i,j}(p_m,\upsilon,\omega) - \beta_{i,j}(q_m,\upsilon,\omega)\right)$$

によって計算される．

5. 3 層ニューラルネットワークを用いた評価関数の性能調査

本研究では，3 層ニューラルネットワークを用いた提出手役評価関数を作成し性能を評価した．教師データには，大貧民の棋譜を図 3 のような盤面データに切り出したもの使用した．

盤面データは以下の情報からなる．

- 場に関する情報
  - 革命の有無（Order）
  - しばりがある場合にはそのスート（Lock_Suits）
  - 最後に場に出された役（Last_Meld）
  - 各プレイヤがそのターンでプレイ可能かどうか（Can_Play）
- カードに関する情報
  - 各プレイヤが持つカードの枚数（Cards_Num）
  - 場の残存カード（Remaining_Cards）
  - Player0（次にプレイするプレイヤ）が持つカード（Player0_Cards）
- 提出手役に関する情報
  - Player0 が実際に出した手役（Put_Meld）

教師データにおける正解の入力は，Player0 が実際に出した手役であり，不正解の入力は，それ以外の合法手である．図 3 であれば正解の入力は H8 であり，不正解の入力は H7,HQ,PASS である．つまりこの盤面データからは 4 個の教師データが作成される．

多数の教師データを用いて，提出手役評価関数の重みを Player0 が実際に出した手役を提出するように調整することで，Player0 のプレイ方策を模倣した提出手役評価関数が作成されることが期待される．

教師データを作成するために 2 種類の大貧民の棋譜を用いた．1 つは「各合法手に対して 500 回のプレイアウトを

表 1　評価項目に使用した盤面情報

盤面情報の種類	評価項目	評価項目数
場の情報	場のオーダ	1
	場の役と提出役のランク差	5
	場の残存カードのランク	41
提出役の情報	手役のランク	17
	手役のタイプ	2
	手役のサイズ	5
	革命が発生するか	1
	しばりが発生するか	1
	8 切りが発生するか	1
	JOKER を含むか	1
プレイヤの情報	自プレイヤの手札のランク	41
	他プレイヤのカード枚数	4

行う原始モンテカルロ法プレイヤ 5 名」の対戦を棋譜にしたものである．もう 1 つは「人間のプレイヤ 1 名と各合法手に対して 500 回のプレイアウトを行う原始モンテカルロ法プレイヤ 4 名」の対戦を棋譜にしたものである．すべての教師データに対して 1 回だけ学習を行うことを 1 イテレーションとし，1,000 イテレーションの学習を行った．提出手役評価関数の中間層数は 15 と 50 の 2 通りを用意した．学習率は初期値を 0.9 とし，1 イテレーションごとに 0.99 を掛け合わせて使用した．3 層ニューラルネットワークの入力層に対応する評価項目は 120 個用意した．評価項目の詳細を表 1 に示す． 評価項目には，実際のゲームにおいてプレイヤから見えている情報だけを用いた．入力層の値は，評価項目が有効になれば 1，そうでなければ 0 とした．

以上の条件で提出手役評価関数の学習を行った．また，重みの初期値によって提出手役評価関数の性能が変動することが考えられるため，初期値が異なるものを 100 個用意しそれぞれ学習した．

性能調査実験では，提出手役評価関数の性能を調査するため「棋譜で提出された手役」と「提出手役評価関数によって得られた評価値が最も大きい手役」がどの程度一致するか調査した．調査に使用する盤面には，学習時に使用していないものを 1,000 盤面用意した．学習時に使用していない盤面に対して調査を行うことで，未知の盤面に対して調査を行ったことになり，実戦時の性能が評価できる．

実験結果を図 4，5 と表 2 に示す．図中と表中では原始モンテカルロ法プレイヤを MCMP と表記する．Ave は平均最善手的中数，Max は最大最善手的中数，Min は最小最善手的中数，Ave のエラーバーは標準偏差における 95%信頼区間を表している．

図 4，5 を見ると，中間層の数に限らず，学習に使用する盤面データを増やすことで提出手役の一致率が上昇していることがわかる．

図 4 の平均最善手的中数を見ると，盤面データ数 15,000 付近で最善手的中数が頭打ちになっており，そのときの最善手一致率はおよそ 69%である．盤面データ数 1,000 の場

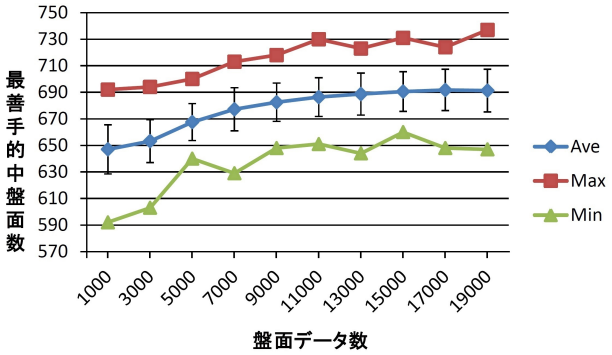

図 4　提出手役一致数（MCMP の棋譜 中間層数 15）

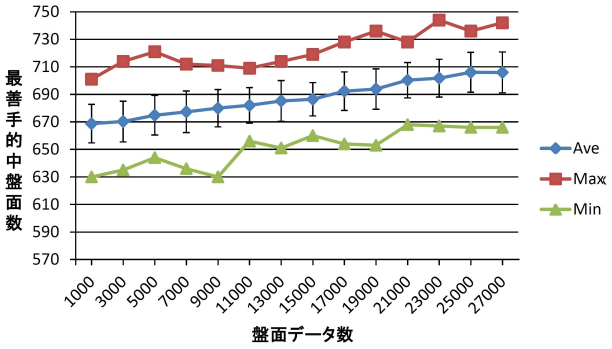

図 5　提出手役一致数（MCMP の棋譜 中間層数 50）

合と比較すると，およそ 4%最善手一致率が向上していることがわかる．

図 5 の平均最善手的中数を見ると，盤面データ数 27,000 まで最善手的中数がゆるやかに上昇し続けており，盤面データ数 27,000 での最善手一致率はおよそ 71%である．中間層数 50 の場合では最善手一致数が頭打ちになっていないため，盤面データ数をさらに増やすことで最善手一致率が上昇する可能性がある．

表 2 を見ると，人間の棋譜を用いた学習は，原始モンテカルロ法プレイヤの棋譜を用いて学習した場合に比べて平均一致率が低くなっている．そのため，人間の棋譜を用いて充分に学習を行うためには，原始モンテカルロ法プレイヤの棋譜よりも多くの棋譜が必要になることがわかる．

6. モンテカルロ評価関数プレイヤの性能調査

本研究では，モンテカルロ法に提出手役評価関数を適用したプレイヤの性能調査を，他のプレイヤとの対戦により行った．

6.1 対戦プレイヤ

対戦に使用するプレイヤは，比較用に作成したプレイヤを 5 つ，UECda ホームページ（プレイヤが入手可能な 2012 年度版）で公開されているプレイヤから 3 つの計 8 プレイヤを使用した．対戦に使用したプレイヤは以下の通りである．

- **Monte**　　原始モンテカルロ法プレイヤ

表 2 提出手役一致数

棋譜	中間層数	盤面データ数	平均	最大	最小	標準偏差
MCMP	15	3,000	653.22	694	603	16.17
MCMP	15	19,000	691.29	737	647	16.07
MCMP	50	3,000	670.27	714	635	14.81
MCMP	50	27,000	705.97	742	666	14.86
人間	15	2,245	420.65	469	362	29.45

- **EVM** 提出手役評価関数プレイヤ（コンピュータの棋譜）
- **EVH** 提出手役評価関数プレイヤ（人間の棋譜）
- **MonteEVM** 提出手役評価関数を適用したモンテカルロ法プレイヤ（コンピュータの棋譜）
- **MonteEVH** 提出手役評価関数を適用したモンテカルロ法プレイヤ（人間の棋譜）
- **Sample** UECda サンプルプレイヤ
- **Nakanaka** UECda ライト級基準プレイヤ
- **paoonR2** 2012 年度 UECda 無差別級優勝プレイヤ

比較用に作成した 5 つのプレイヤの詳細を以下に示す．

Monte

プレイアウト中はパス以外の合法手の中から等確率に選択した手役を提出するようにした．プレイアウト回数は全体で 1,000 回とし，プレイアウトを行う手役は UCB1 値によって決定した．場に提出する手役は，1,000 回のプレイアウト終了後に最も評価値の高い手役である．

EVM

教師データの作成には，各合法手に対して 500 回のプレイアウトを行う原始モンテカルロ法プレイヤ 5 名の対戦の棋譜を用いた．提出手役評価関数は，中間層数が 15，盤面データ数が 19,000，イテレーション数が 1,000 のものである．場に提出する手役は，提出手役評価関数によって得られる評価値が最も大きい手役である．

EVH

教師データの作成には，人間のプレイヤ 1 名と，各合法手に対して 500 回のプレイアウトを行う原始モンテカルロ法プレイヤ 4 名の対戦の棋譜を用い，人間のプレイヤのプレイ方策を学習した．提出手役評価関数は，中間層数が 15，盤面データ数が 3,245，イテレーション数が 1,000 のものである．場に提出する手役は，提出手役評価関数によって得られる評価値が最も大きい手役である．

MonteEVM

教師データの作成には，各合法手に対して 500 回のプレイアウトを行う原始モンテカルロ法プレイヤ 5 名の対戦の棋譜を用いた．プレイアウトに用いる提出手役評価関数は，中間層数が 15，盤面データ数が 19,000，イテレーション数が 1,000 のものを使用し，プレイアウト中は提出評価関数によって得られる評価値が最も大きい手役を提出するようにした．プレイアウト回数，プレイアウトを行う手役の決定手法，場に提出する手役は Monte と同様である．

MonteEVH

教師データの作成には，人間のプレイヤ 1 名と，各合法手に対して 500 回のプレイアウトを行う原始モンテカルロ法プレイヤ 4 名の対戦の棋譜を用い，人間のプレイヤのプレイ方策を学習した．プレイアウトに用いる提出手役評価関数は，中間層数が 15，盤面データ数が 3,245，イテレーション数が 1,000 のものを使用し，プレイアウト中は提出評価関数によって得られる評価値が最も大きい手役を提出するようにした．プレイアウト回数，プレイアウトを行う手役の決定手法，場に提出する手役は Monte と同様である．

MonteEVM, MonteEVH はプレイアウト中でも合法手にパスを含むようにした．Monte, EVM, EVH, MonteEVM, MonteEVH の手札交換のアルゴリズムは Nakanaka と同じものを使用した．

6.2 対戦実験

対戦実験では，比較用に作成した 5 つのプレイヤと UECda ライト級基準プレイヤの Nakanaka を対戦させた．対戦は 5 人対戦で行い，その組み合わせは，「比較するプレイヤの数」と「Nakanaka の数」が 1 対 1 もしくは 2 対 2 となるようにし，残りのプレイヤは Sample とした．

対戦結果を表 3，4 に示す．表中の点数は，平民を 0 点として，大富豪を +2 点，富豪を +1 点，大貧民を −2 点，貧民を −1 点としたときの総獲得点数である．試合数は 2012 年度の大会試合数と同じ 400 である．

表 3，4 を見ると，いずれの対戦でも Monte，EVM，EVH は Nakanaka に獲得点数で負けている．一方，表 3 の MonteEVM と Nakanaka の対戦結果を見ると，MonteEVM は Nakanaka に獲得点数で勝っている．この結果から，モンテカルロ法に提出手役評価関数を適用することでプレイヤが強化されたことがわかる．

また参考として，対 Nakanaka 戦において最も獲得点数が多い MonteEVM と 2012 年度 UECda 優勝プログラムの paoonR2 を対戦させた．対戦結果を表 5，6 に示す．

表 5，6 を見ると，どちらの対戦においても MonteEVM は Nakanaka と paoonR2 に獲得点数で負けている．また，

表 3 1 対 1 の対戦結果（VS Nakanaka）

プレイヤ名	プレイヤ	Nakanaka	Sample	Sample	Sample	プレイヤと Nakanaka の差
Monte	+182	+395	−134	−168	−275	−213
EVM	−191	+468	−87	−120	−70	−659
EVH	−537	+434	+86	+42	−25	−971
MonteEVM	+346	+239	−154	−190	−241	+107
MonteEVH	−18	+428	−56	−173	−181	−446

表 4 2 対 2 の対戦結果（VS Nakanaka）

プレイヤ名	プレイヤ	プレイヤ	Nakanaka	Nakanaka	Sample	プレイヤと Nakanaka の差
Monte	+24	−23	+157	+116	−274	−272
EVM	−284	−306	+430	+312	−152	−1332
EVH	−468	−479	+463	+396	+88	−1806
MonteEVM	+85	−56	+240	+99	−368	−310
MonteEVH	−102	−223	+323	+225	−223	−873

表 5 1 対 1 の対戦結果（VS paoonR2）

プレイヤ名	プレイヤ	paoonR2	Sample	Sample	Sample
MonteEVM	+243	+533	−202	−273	−301

表 6 2 対 2 の対戦結果（VS paoonR2）

プレイヤ名	プレイヤ	MonteEVM	paoonR2	paoonR2	Sample
MonteEVM	−61	−61	+314	+290	−482

表 4 の MonteEVM と Nakanaka の対戦を見ると，1 対 1 の対戦では MonteEVM の方が獲得点数が多かったが，2 対 2 の対戦では Nakanaka の方が獲得点数が多くなっている．

表 3，4 の MonteEVM について見ると，Monte よりも獲得点数が少なくなっているため，人間の棋譜から学習した提出手役評価関数をモンテカルロ法に適用してもプレイヤが強化されていないことが分かる．これは盤面データ数が 3,245 と少ないことや，盤面データの中に人間が明らかなミスをしたものも含まれている可能性があることなどが原因であると考える．人間の棋譜から学習した提出手役評価関数の性能は，棋譜を充実化することで改善できると考える．

7. まとめ

本研究では，3 層ニューラルネットワークを用いた提出手役評価関数の性能調査を行い，モンテカルロ法プレイヤのプレイアウト部分に提出手役評価関数を適用することでプレイヤの強化を図った．

提出手役評価関数の性能を評価した結果，中間層数 15 の場合では，学習に使用する盤面データを増やすことで提出手役の一致率が上昇し，盤面データ数 15,000 程度で充分に学習できていることが確認できた．盤面データ数 15,000 の提出手役評価関数では，未知の盤面に対する提出手役一致率がおよそ 69%となった．また，中間層数 50 の場合でも同様に，盤面データを増やすことで提出手役の一致率が上昇した．盤面データ数 27,000 までの提出手役一致率を調査したが，一致率が頭打ちになっておらず，充分な学習にはこれ以上の盤面データが必要であることが確認できた．対戦実験では，2012 年度のコンピュータ大貧民大会優勝プレイヤには勝てなかったが，モンテカルロ法に対して提出手役評価関数を適用することでプレイヤが強化された．

今後の課題として，提出手役評価関数の改良と，棋譜の充実化が挙げられる．提出手役評価関数の改良では，評価項目の数を増やす，評価項目の設計を改良する，序盤・中盤・終盤で使用する評価関数を分ける，などの方法が考えられる．また，棋譜を充実化させることで，学習によって得られたプレイヤの棋譜による再学習や，強い人間の棋譜による学習などが行えるようになる．充実化した棋譜を用いて，改良された提出手役評価関数の学習を行うことで，強いプレイヤを作り出すことができると考える．

参考文献

[1] M. Buro. Improving heuristic mini-max search by supervised learning. Artificial Intelligence, Artificial Intelligence 134(1–2), pp. 85–99 (2002).
[2] P. Auer, N. Cesa-Bianchi and P. Fischer. Finite-time Analysis of the Multi-armed Bandit problem. *Machine Learning*, Vol. 47, pp. 235–256 (2002).
[3] 伊藤 祥平, 但馬 康宏, 菊井 玄一郎. コンピュータ大貧

民における高速な相手モデル作成と精度向上. **数理モデル化と問題解決研究会報告**, Vol. 2013-MPS-96, No. 4, pp.1–3 (2013).

[4] 金子 知適. 兄弟節点の比較に基づく評価関数の調整. **第 *12* 回ゲームプログラミングワークショップ**, pp. 9–16 (2007).

[5] 金子 知適, 田中 哲朗, 山口 和紀, 川合 慧. 駒の関係を利用した将棋の評価関数. **第 *8* 回ゲームプログラミングワークショップ**, pp. 14–21 (2003).

[6] 金子 知適, 山口 和紀. 将棋の棋譜を利用した，大規模な評価関数の調整. **第 *13* 回ゲームプログラミングワークショップ**, pp. 152–159 (2008).

[7] 北川 竜平, 三輪 誠, 近山 隆. 麻雀の牌譜からの打ち手評価関数の学習. **第 *12* 回ゲームプログラミングワークショップ**, pp. 76–83 (2007).

[8] 小沼 啓, 西野 哲朗. コンピュータ大貧民に対するモンテカルロ法の適用. **研究報告ゲーム情報学** *(GI)*, Vol. 2011-GI-25, No. 3, pp.1–4 (2010).

[9] 須藤 郁弥, 成澤 和志, 篠原 歩. UEC コンピュータ大貧民大会向けクライアント「snowl」の開発. 第 2 回 UEC コンピュータ大貧民シンポジウム (2011).

[10] 須藤 郁弥, 篠原 歩. モンテカルロ法を用いたコンピュータ大貧民の思考ルーチン設計. 第 1 回 UEC コンピュータ大貧民シンポジウム (2010).

[11] 保木 邦人. 局面評価の学習を目指した探索結果の最適制御. **第 *11* 回ゲームプログラミングワークショップ**, pp. 78–83 (2006).

[12] 竹内 聖悟. コンピュータ将棋の技術と GPS 将棋について. `http://www.cybernet.co.jp/avs/documents/pdf/seminar\_event/conf/19/1-3.pdf` (2013).

[13] 電気通信大学. UEC コンピュータ大貧民大会, `http://uecda.nishino-lab.jp/2012/` (2012).