札譜データの学習を用いた 大貧民モンテカルロプレイヤの強化

岡 和人[1,a)]　松崎 公紀[1,b)]

概要：近年，コンピュータ大貧民において，乱数によるシミュレーションを用いたモンテカルロ法プレイヤが広く用いられている．モンテカルロ法プレイヤでは各盤面から各プレイヤの手を乱数を用いてシミュレーションを行う（このシミュレーションをプレイアウトと呼ぶ）が，このプレイアウトの確度を向上することでより強いモンテカルロ法プレイヤを実現することができる．これまでに著者らは，札譜データを用いて，各盤面においてよく出される手の学習を行った．本研究では，その学習の結果を用いて，プレイアウトの確度の向上を試みる．本発表では，複数のプレイヤの札譜から学習した合法手の順位付けを用いて，どのようにモンテカルロ法プレイヤの強さが向上するかについて報告する．

キーワード：大貧民，機械学習，モンテカルロ法

1. はじめに

本研究の対象とする大貧民（大富豪とも呼ばれる）は，トランプを使った多人数不完全情報ゲームの 1 つである．大貧民をコンピュータにプレイさせるコンピュータ大貧民は，2006 年に電気通信大学での大会 UECda [13] が始まって以降，近年活発に研究が行われるようになっている．その中でも，乱数によるシミュレーションを行うことで着手選択を行うモンテカルロ法は標準的な手法である．実際，UECda では 2009 年以降モンテカルロ法またはその応用を用いたプレイヤが優勝している．ゲームに対するモンテカルロ法において，ある盤面から終局までを乱数によるシミュレーションによってプレイすることをプレイアウトと呼ぶ．大貧民では手札の枚数が単調に減少するため，プレイアウトの実装が容易であり，そのことは大貧民においてモンテカルロ法が適している理由の 1 つである．

一般に，モンテカルロ法によってより正確な解を得るためには，シミュレーションの精度と確度を高めることが重要である．したがって，モンテカルロ法プレイヤの強さ向上には，プレイアウトの精度と確度を向上することが重要である．しかし，単純なモンテカルロ法で用いられる，乱数を用いて等確率に着手選択する方法では，プレイアウトの確度を十分に得られないことがある．この原因の 1 つは，実際のゲームにおいては選択されないような弱い手を選択してしまうためである．

例えば，あるプレイアウトの途中で，図 1 に示される状況があったとする[*1]．ここで，残っているプレイヤは自分を含めて 2 人であるので，すべての残りカードが分かっている．図 1 の盤面にお

[1] 高知工科大学
[a)] 150293s@ugs.kochi-tech.ac.jp
[b)] matsuzaki.kiminori@kochi-tech.ac.jp

[*1] 大貧民のルールについては，第 2.1 節を参照

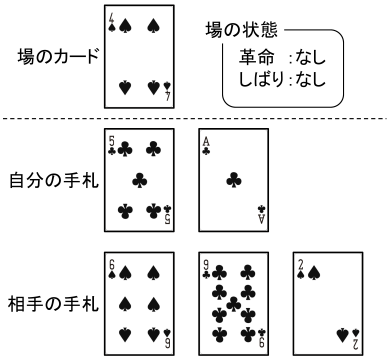

図 1 プレイアウト確度が低下する局面の例

いて，自分が「♣5」を出せば，その後相手がどのようにカードを出しても自分が勝つことができる．一方，自分が「♣A」を出すと，その後相手が「♠2」を出すことで，どのようにしても相手が勝つ．この盤面でのプレイアウトの結果は，「♣5」という手で勝てるので「勝ち」であるべきである．しかし，（出せる手札があるときはパスすることなく）ランダムに着手選択すると，0.5 の確率で「勝ち」，0.5 の確率で「負け」となってしまう．これは，モンテカルロ法でランダムにプレイアウトする限り避けられない問題である．

このような，モンテカルロ法におけるプレイアウトの確度の問題を解決する 1 つの方法は，モンテカルロ法に木探索の要素を加えたモンテカルロ木探索 [4] である．モンテカルロ木探索ははじめにコンピュータ囲碁において提案され，コンピュータ囲碁プレイヤの強さを大幅に向上させた [16]．その後，モンテカルロ木探索アルゴリズムを他のゲームに適用する研究が多く行われている．しかし，大貧民は相手の手札が見えない不完全情報ゲームであるため，モンテカルロ木探索を適用することはこれまでうまくいっていない．

本研究では，機械学習の手法を応用することによって，モンテカルロ法におけるプレイアウトの確度を向上させることを目標とする．機械学習をゲームプレイヤの評価関数に適用する考え方は比較的古くからある [2], [5]．近年，コンピュータ将棋においては，機械学習によって多数の棋譜データから評価関数のパラメータを調整した Bonanza [15] 以降，多くのコンピュータ将棋プレイヤにおいて機械学習の手法が利用されている．実際，機械学習によって強化されたコンピュータプレイヤが，将棋のプロ棋士に勝つなど成果を挙げている．

本研究では，まず，多数の札譜（将棋などにおける棋譜）データから学習を行い，ある盤面においてある手役を出すべきかどうかを計算する提出手役評価関数を作成する．その上で，作成した提出手役評価関数を用いて，プレイアウト中の盤面における手役の優先順位を決定する．例えば，図 1 の盤面において，「♣A」よりも「♣5」を出すべきであると計算する．プレイアウトにおいて優先順位の高い手を出すことにより，プレイアウトが実際のゲームに近づき，したがってプレイアウトの確度が向上することが期待される．

本研究では，提出手役評価関数を学習するアルゴリズムには，先行研究 [8], [12] で用いられた 3 層ニューラルネットワークと，アルゴリズムや学習が単純である平均化パーセプトロン [3] を用いる．また，学習に用いる教師データには，4 種類のプレイヤから得た札譜データを用いる．学習によって得られ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，単純モンテカルロ法やその他のプレイヤと比較することにより，提案手法の効果を調べる．

本論文の貢献はおおきく以下の 3 つである．

- 4 つの異なるプレイヤの札譜からそれぞれ学習を行い，提出手役評価関数が求めた最善手と札譜データにおいて出された手が一致する確率を調査する（第 5 章）．
- 学習によって得られた提出手役評価関数を用いて，モンテカルロ法のプレイアウトを改善する手法を提案する（第 4 章）．
- 提案手法プレイヤと既存のプレイヤの対戦を行い，その強さを評価する．また，教師データに用いた札譜データと，提案手法によるプレイヤの強化との関連を調査する（第 6 章）．

本論文の構成は以下のとおりである．第 2 章では，準備として大貧民のルールと既存のプレイヤプログラムを導入する．第 3 章では，2 つの学習アルゴリズムと，その大貧民への適用について示す．第 4 章では，学習によって得られた提出手役評価

関数を用いて，モンテカルロ法のプレイアウトを改善する方法を示す．第 5 章では，第 6 章で行う実験の準備のため，学習によって得られた提出手役評価関数の性能と，プレイアウト中の手役選択に用いるパラメータを調査する．第 6 章では，提案手法によるプレイヤの強化について，実験により評価・考察する．第 7 章で関連研究を示し，第 8 章で本論文のまとめと今後の課題を述べる．

2. 準備

2.1 大貧民

大貧民（大富豪とも呼ばれる）は，多人数で行うトランプゲームである．最初に配られる手札からルールに沿って 1 枚または複数枚のカードを役として場に出していき，手札がなくなる順位を競う．本研究では，コンピュータ大貧民大会 UECda の標準ルール 2010 年版 [13] に従う．ただし，説明を簡潔にするため，後述する相対得点を用いる．以下に，本論文に関係する重要なルールを示す．

人数 ゲームは 5 つのプレイヤで行う．

ランク カードは 3 が一番弱く，数字が大きくなるほど強くなる．A は K より強く，2 は A より強い．

カードの出し方 カードの出し方には単体役，複数役，階段役の 3 種類がある．場と同じ種類・枚数の役で，より強いランクの役を出せる．

複数役 同じランクのカードを 2 枚以上で出す役を複数役と呼ぶ．

階段役 同じスートでランクが連続するカードを 3 枚以上で出す役を階段役と呼ぶ．

ジョーカー ジョーカーは後述するスペ 3 切りを除き，最強のカードとして扱われる．複数役と階段役では，任意のカードの代わりと出来る．

パス 自分の手番では，役を出すかパスをすることを選択する．パスをした場合，場が流れるまで自分の手番は来ない．

場の流れ 全てのプレイヤがパスをすると場が流れる．場が流れると，最後に場に役を出したプレイヤが次に任意の役を出す権利を持つ．

革命 4 枚以上の複数役か，5 枚以上の階段役が出されたとき，革命が起こる．革命が起こると，カードの強さが逆転する．革命は，ゲームが終了するか再び革命が起こるまで続く．

スペ 3 切り 場にジョーカーが単体役で出されている場合，「♠3」を単体役で出すことが出来る．その後，場は流れる．

8 切り 役に 8 のランクが含まれると 8 切りが起こり，場が流れる．

しばり 場のカードと同じスートでカードが出されるとしばりとなり，場のカードと完全に同じスートの役しか出せなくなる．

上がり 手札が無くなると上がりとなる．上がりの際も任意の手役を出せる．

得点 各ゲームで最初に上がったプレイヤから順に，2，1，0，−1，−2 点を得る．

手札交換 前ゲームの順位によって，次ゲームの手札配布後に，以下のように，手札交換を行う．

- 1 位は 5 位に好きなカードを 2 枚渡す
- 2 位は 4 位に好きなカードを 1 枚渡す
- 4 位は 2 位に最も強いカードを渡す
- 5 位は 1 位に強いカードから順に 2 枚渡す

本研究では，UECda で公開されている標準 C サーバを用いて対戦を行う．各プレイヤの組み合わせについて，ゲーム数は 10000 とした．なお，3 ゲーム毎にプレイヤの席順が変更され，100 ゲーム毎に手札交換のない初期状態でゲームが始まる．

2.2 プレイヤ

提出手役評価関数の学習に用いる札譜データを取得するために，以下の 4 プレイヤを用いる．また，評価実験においてもこれらのプレイヤを利用する．以下のプレイヤのうち，paoonR2 と kishimen は UECda ウェブサイトで公開されている[*2]．

paoonR2 2012 年度 UECda の，無差別級部門（モンテカルロ法や学習を用いるプレイヤからなる）で優勝したプレイヤである．

kishimen 2013 年度 UECda の，ライト級部門（ルールベース，もしくはそれと同程度の計算量で手を出すプレイヤからなる）で優勝したプレイヤである．

[*2] http://uecda.nishino-lab.jp/2014/download.php

表 1 既存プレイヤの対戦結果

プレイヤ	paoonR2	kishimen	MC	Greedy
相対得点	+11459	+5314	+2039	−9406

MC 自作した単純なモンテカルロ法プレイヤである．提出手役または交換手札を求めるためのプレイアウト時には，パスを含まない合法手からランダムに着手を選択する．アルゴリズムの詳細は後述する．

Greedy 合法手のうち，一番枚数が多い役を出す．最大枚数の役が複数あるときは，その中で一番ランクの低い役を出す．手札交換では，ランクの低いカードを渡す．

各プレイヤの強さの指標として，paoonR2, kishimen，MC，および Greedy 2 つで対戦を行い，それぞれが得た得点を表 1 に示す．プレイヤ MC は，中程度の強さであり，kishimen よりも少し弱い．

プレイヤ MC のアルゴリズムの詳細

プレイヤ MC は，単純なプレイアウトによって提出手役と交換手札を選定する．提出手役の選択を行うアルゴリズムを図 2 に，プレイアウトのアルゴリズムを図 3 にそれぞれ示す．

自分の手番では，まず合法手を列挙する．この合法手には，場が新しくなければパスを含む．合法手が 1 つしかない場合，または，出せば必ずすぐに勝てるような合法手がある場合には，その手を選択する．そうでなければ，規定回数のプレイアウトを行い，このプレイアウトで得られた得点の平均が最も高い手を提出手役とする．ある手番において手を選択するために行うプレイアウトの回数は，札譜を取るとき（第 5.1 節）では 3000 回，対戦実験（第 6 章）では 1500 回とした．各プレイアウトの最初では，合法手の中で UCB1 値が最大となるものを提出手役とする．UCB1 値を求める手を j とする．その手で得られた得点の平均を $\overline{X_j}$，全ての手で行われたプレイアウト回数の合計を n，手 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 とすると，UCB1 値は

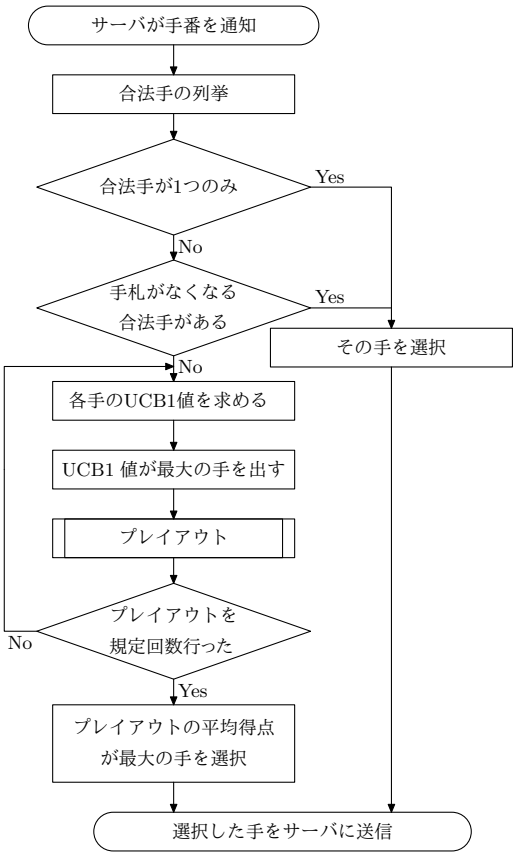

図 2 モンテカルロ法プレイヤ（MC）の手選択

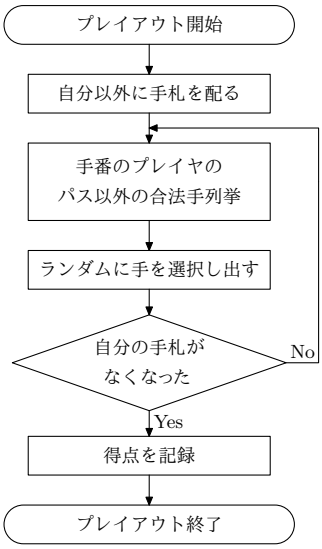

図 3 モンテカルロ法プレイヤ（MC）のプレイアウト

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

で与えられる．バランスパラメータ c は，1 ゲームの得点の最大と最小の差から 4 とした．

各プレイアウトは次の手順で行う．まず，プレイアウトの始めに，自分以外のプレイヤに対して手札を分配する．手札交換の影響を考慮して，残っているカードから強い順に最大 6 枚のカードを，1 位のプレイヤには 3 枚，2 位のプレイヤには 2 枚，3 位のプレイヤには 1 枚，それぞれランダムに分配する．その後残ったカードをランダムに分配する．次に，自分の手札がなくなるまで，着手選択と仮想ゲームの進行を繰り返す．ここで，プレイアウトの内部では，パス以外の合法手の中からランダムに着手を選択する．

各ゲームの始めの手札交換では，モンテカルロ法を用いて渡すカードを選択する．それぞれのカードについて，それを交換したと仮定してプレイアウトを行い，得られた平均得点が最も高くなるカードを相手に渡すカードとする．

3. 提出手役評価関数

本研究では，出す手役の優先度を求める評価関数（提出手役評価関数）を機械学習を用いて作成する．提出手役評価関数は，場と手札の状態，評価値を計算する手を入力として与えることで，その手を選択するべきかどうかを評価値として出力する．機械学習の手法には，3 層ニューラルネットワーク（NN）と平均化パーセプトロン（AP）の 2 種類を用いる．本章では，これらの機械学習の手法の概要と，ある盤面における手を評価するための評価項目の要素について示す．

3.1 3 層ニューラルネットワーク

図 4 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 $\omega_{i,j}$ とし，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．関数 $\sigma(x)$ をシグモイド関数

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

とする．このとき，各層の計算は以下のように行われる．まず，入力層の各ユニットは与えられた入力 p から値を計算する．入力層のユニット i で計算される値を $f_i(p)$ と書く．次に，中間層の各ユニットは入力層の出力を入力として値を計算する．中間層のユニット j で計算される値 $g_j(p)$ は

$$g_j(p) = \sigma\Big(\sum_i \omega_{i,j} f_i(p)\Big)$$

である．最後に，出力層のユニットは，中間層の出力を入力として値を計算する．3 層ニューラルネットワークの出力値 $h(p)$ は

$$h(p) = \sigma\Big(\sum_j v_j g_j(p)\Big)$$

となる．

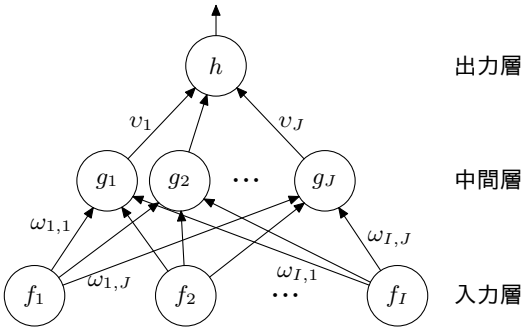

図 4 3 層ニューラルネットワーク

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．大貧民の札譜データのある盤面において，学習対象のプレイヤが出した手役を正解，学習対象のプレイヤが出さなかった合法手を不正解とする．ある盤面における不正解 p_m と正解 q_m の組を，1 つの教師データとする．よって，1 つの盤面から，合法手の数から 1 を引いた数の教師データが得られる．誤差逆伝播法による重み調整では，不正解の出力と正解の出力の差

$$E = \sum_m \sigma\left(h(p_m) - h(q_m)\right)$$

を最小化することを目標とする．

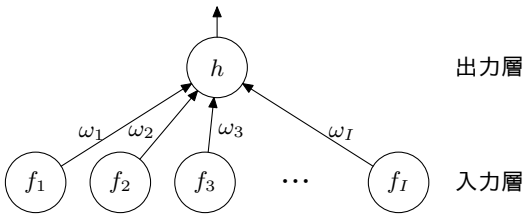

図 5 平均化パーセプトロン

3.2 平均化パーセプトロン

平均化パーセプトロン [3] は，計算や学習が容易であり，また学習した判別器の性能も比較的高いことから広く用いられている機械学習手法である．平均化パーセプトロンの構成を図 5 に示す．

入力層のユニット i に対応する重みを ω_i とする．このとき，以下のようにして値が計算される．まず，入力層の各ユニットは，与えられた入力 p から値 $f_i(p)$ を計算する．次に，平均化パーセプトロンの出力値 $h(p)$ は，入力層の各ユニットの値の重み和

$$h(p) = \sum_i \omega_i f_i(p)$$

によって得られる．

平均化パーセプトロンの学習で用いる教師データは，3 層ニューラルネットワークの学習で用いる教師データと同一である．平均化パーセプトロンの学習は以下のようにして行う．教師データ m 組の学習によって得られた重みを ω_i^m とする．重みの初期値は $\omega_i^0 = 0$ とし，各教師データについて以下のように重みを更新する．

$$\omega_i^m = \begin{cases} \omega_i^{m-1} & (h(p_m) > h(q_m) \text{ のとき}) \\ \omega_i^{m-1} + f_i(p_m) - f_i(q_m) & \\ & (h(p_m) \leq h(q_m) \text{ のとき}) \end{cases}$$

このようにして計算された重み ω_i^m を教師データ m について平均をとり，最終的な重みとする．

3.3 入力層に用いる評価項目

3 層ニューラルネットワークおよび平均化パーセプトロンでは，問題に対して適切な入力層を与える必要がある．本研究では，実際のゲームにおいてプレイヤが知り得る情報，すなわち，場の状況，自分の手札，および，提出手役によって計算できる情報を入力層のための評価項目とする．各評価項目は，1（真）または 0（偽）の値をとる．評価項目の詳細を表 2 に示す．

3 層ニューラルネットワークでは，表 2 に与えられる評価項目をそのまま入力層とした．よって，3 層ニューラルネットワークの入力層のユニット数は 182 である．平均化パーセプトロンでは，表 2 に与えられる評価項目から，手役の情報であるグループ B と手以外の盤面についての情報であるグループ C の直積を入力層とした．よって平均化パーセプトロンの入力層のユニット数は $146 \times 35 = 5110$ である[*3]．

4. 学習した評価関数のモンテカルロ法への適用

本研究の主な目的は，札譜データより学習した評価関数を用いて，モンテカルロ法のプレイアウトの確度を向上させることである．本章では，札譜から学習した評価関数をどのようにプレイアウトで用いるのかを説明する．

第 3 章で示したとおり，評価関数の学習は，合法手のうち出された手役（正解）と出されなかった手役（不正解）の差を最小化するように行った．したがって，評価関数の出力の絶対的な値には意味がなく，評価関数は手の順序を定めることにしか用いることができないことに注意が必要である．

そこで，プレイアウト中のある盤面において，出す手の選択は次のようにして行った（図 6）．始めに，その盤面での合法手を列挙する．ここで，場が新しい場合を除き，パスも合法手に含むようにした．次に，すべての合法手について，その評価値を計算する．求めた評価値に基づき，評価値が降順となるよう合法手を並び換える．最後に，乱数によってその盤面において出す手を選択する．ここで，評価値の高い手の方が，より大きな確率で選択されるようにした．具体的には，パラメータ α（$0 \leq \alpha \leq 1$）を用いて，i 番目の手が選択される確率 p_i と $i+1$ 番目の手が選択される確率 p_{i+1} の間

[*3] このうち，学習の結果用いられなかった（対応する重みが 0 である）入力層のユニットが 442 ある．

表 2 入力層に用いる評価項目

グループ	要素数	評価基準
A	1	バイアス項（常に 1 とする）
B	5	「場に出ているランク」と「場に出した役のランク」の差
	3	出した役の枚数 (2, 3, 4 枚以上)
	17	出した役のランク *1
	2	革命を起こしたか
	2	しばりを起こしたか
	2	8 切りを起こしたか
	4	役の種類 (パス，単体役，複数役，階段役)
C	2	役を出す前に革命状態であるかどうか
	41	場に出ておらず自分も持っていないカードの，ランクごとの枚数 ∗2
	41	役を出した後の手札の，ランクごとの枚数 ∗1
	5	勝ち抜けまたはパスしたプレイヤの数 (0, 1, 2, 3, 4)
	8	自分以外の各プレイヤがパスしているかどうか
	6	自分の手札枚数 (1, 2, 3, 4–5, 6–8, 9 枚以上)
	28	自分以外の 4 プレイヤの手札枚数を少ない順に (0, 1, 2, 3, 4–5, 6–8, 9 枚以上)
	13	場の役のランク (3, 4, . . ., K, A, 2)
	2	場が新しいかどうか

*1 3 から 2 までの 13 のランクに加えて，単体役のジョーカーとスペ 3 切り（それぞれ通常時と革命時）
*2 13 の各ランク毎に 0 枚，1 枚，2 枚以上の 3 通り．加えて，ジョーカーの有無 ($3 \times 13 + 2 = 41$).

に，関係式 $p_{i+1}/p_i = \alpha$ が成り立つものとする[*4]．ただし，パラメータ $\alpha = 0$ のとき評価値が最大となる手を決定的に選択し，パラメータ $\alpha = 1$ のとき完全にランダムに手を選択することとなる．評価実験においては $\alpha = 0.6$ としたが，その詳細については第 5.2 節にて述べる．

5. 実験準備

5.1 札譜データの学習

教師データである札譜から学習を行い，提出手役評価関数を作成し，性能を調査する．

学習で用いる札譜データは，第 2.2 節で述べた 4 つのプレイヤ（Greedy，kishimen，paoonR2，MC）より作った．それぞれのプレイヤについて，同一プレイヤのみからなるゲームを行い，60000 盤面分の教師データを得た．ただし，この教師データには，ある手を出すことで自分の手札が無くなるような盤面と，パス以外に合法手が無い盤面は含まれない．

[*4] 例えば，合法手の数が 4，パラメータ $\alpha = 0.5$ のとき，手が選択される確率はそれぞれ $p_1 = 0.533$，$p_2 = 0.266$，$p_3 = 0.133$，$p_4 = 0.066$ となる．

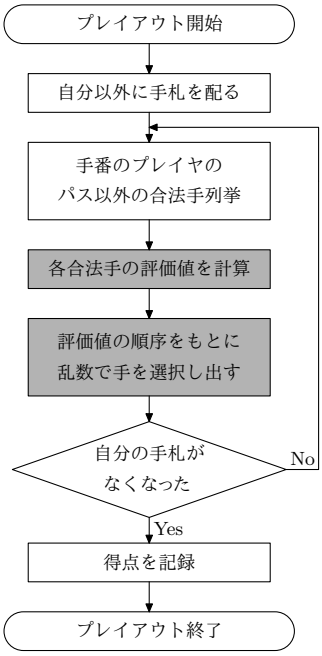

図 6 学習した提出手役評価関数を適用したプレイアウトアルゴリズム．色付けは，単純なプレイアウトとの相違点を表す．

表 3 評価関数の学習結果

	NN				AP			
学習盤面数	paoonR2	kishimen	MC	Greedy	paoonR2	kishimen	MC	Greedy
1000	0.659	0.811	0.772	0.833	0.655	0.825	0.746	0.892
5000	0.669	0.825	0.779	0.865	0.684	0.868	0.767	0.931
10000	0.667	0.826	0.779	0.873	0.695	0.882	0.776	0.939
30000	0.658	0.830	0.778	0.888	0.710	0.900	0.787	0.945
50000	0.649	0.832	0.800	0.899	0.714	0.903	0.792	0.947

表 4 最大一致率とそのときの盤面数

学習	プレイヤ	最大一致率	その盤面数
NN	paoonR2	0.669	6000
	kishimen	0.833	45000
	MC	0.781	49000
	Greedy	0.899	49000
AP	paoonR2	0.714	50000
	kishimen	0.903	50000
	MC	0.793	50000
	Greedy	0.947	50000

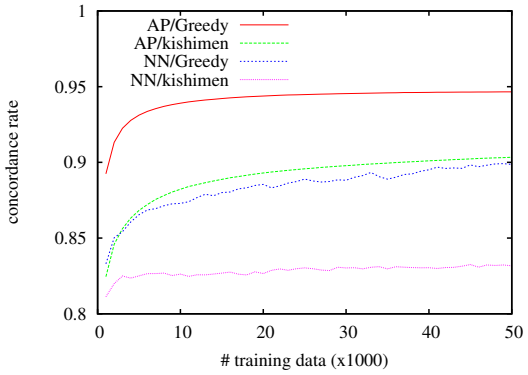

図 7 Greedy と kishimen の札譜による学習結果

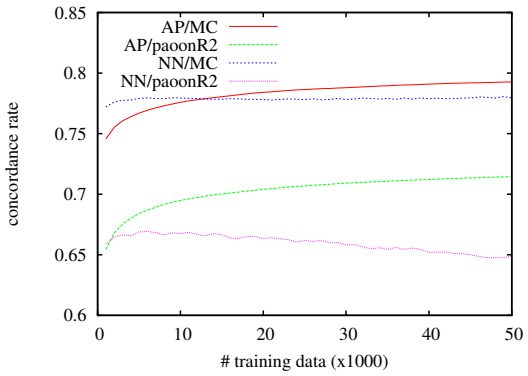

図 8 MC と paoonR2 の札譜による学習結果

これらの 60000 盤面から，最大 50000 盤面をランダムに選択して 3 層ニューラルネットワークおよび平均化パーセプトロンによってそれぞれ学習を行う．学習は，1000 盤面学習する毎に学習過程として重みを書き出した．学習の繰り返し回数は 1 とした[*5]．学習に用いなかった盤面のうち 10000 盤面を評価盤面とし，合法手のうち，提出手役評価関数で求めた評価値が最大となる手と，札譜データにおいて出された手が一致するかどうかを調べた．これらの実験を，それぞれ乱数の種を変えて 100 回行い，提出手役の一致率の平均を比較した．

学習を行った提出手役評価関数の提出手役一致率の一部を表 3 に，得られた結果を全てプロットしたものを図 7 及び図 8 に示す．また，提出手役一致率が最大となる学習盤面数とそのときの一致率を表 4 に示す．

提出手役の一致率は，最も高い Greedy の札譜からの学習では 89.9〜94.7%，最も低い paoonR2 の札譜からの学習では 66.9〜71.4%であった．この結果は，これまでに示されている伊藤ら [6] の結果（snowl に対して約 4 割）や地曳ら [12] の結果（単純なモンテカルロ法プレイヤに対して約 7 割）に比べて高い一致率となった．第 3.3 節で示した入力層の評価項目が既存手法よりも場の情報をより多く持つことがその理由であると考える．また，学習盤面数が 1000[*6]の時点でも，提出手役評価関数は，50000 盤面の学習での一致率の 91%以上の一致率を示している．これは，実際のゲーム中で

[*5] 繰り返し回数を増やしても，学習結果はほとんど良くならない，もしくは悪化することが見られた．

[*6] 1 つのプレイヤが，対戦を 150 ゲーム行った札譜から，盤面データをおよそ 1000 個作成することが出来る．

オンライン学習することの可能性を示している．学習アルゴリズム間の違いとして，平均化パーセプトロンでは学習盤面数を増やすと一致率が向上するのに対し，3 層ニューラルネットワークではあまり向上せず，特に paoonR2 の札譜からの学習では一致率が減少してしまうという結果となった．

以降の実験では，表 4 に示す評価関数を用いる．

5.2 手選択における確率の調査

提出手役評価関数を提案手法のプレイヤの手選択とカード交換に用いる際に必要となる，手を選択する確率が変化する比率 α を決定するため，α を変動させて対戦を行い，プレイヤの強さの変化を調査する．対戦は，3 層ニューラルネットワークを用いて学習を行った提案手法のプレイヤ 1 つに対し，残りの 4 プレイヤを MC とした組み合わせで行う．

提案手法のプレイヤが対戦で得られた得点をプロットしたものを図 9 に示す．$\alpha = 1$ のときは，評価値によらず均等に手が選ばれるため，単純なモンテカルロ法プレイヤと同じ動作である．実際，結果では，相対得点がほぼ 0 となっている．一方，$\alpha = 0$ のときは，各盤面において評価値の最も高い手を選ぶ．このときは，どの札譜データから学習した評価関数を使った場合でも，元のモンテカルロプレイヤよりも大幅に弱いという結果となった．これは，乱数による多数のプレイアウトを行うことで得られるモンテカルロ法のメリットが失われたことが大きな原因であると考える．相対得点が最も高くなったのは，$0.6 \leq \alpha \leq 0.7$ のときであった．例えば，$\alpha = 0.6$ とすると，最も評価値の高い手が選ばれる確率はおよそ 0.4 しかなく，また上位 3 つの手の確率の合計もおよそ 0.8 しかない[*7]．このように，評価値の高い手が選ばれる確率が小さい場合，言い変えると，評価値の低い手も選ばれる確率が高い場合により強いプレイヤとなったことは，著者らの想像とは異なる結果であった．

以降の第 6 章の実験においては，提案手法プレイヤでは $\alpha = 0.6$ とする．

[*7] 合法手の数が少なければ，これらの確率はもう少し大きくなる．

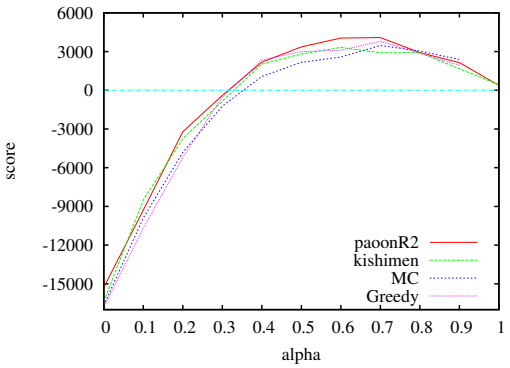

図 9 手選択における確率と強さ

表 5 プレイヤ MC との対戦結果

	相対点数	
学習データ	NN	AP
paoonR2	+4447	+3285
kishimen	+3364	+4051
MC	+3822	+3997
Greedy	+3782	+2966

6. 対戦による評価実験

6.1 モンテカルロプレイヤの強化についての評価

札譜データから学習した提出手役評価関数を用いることで，プレイアウトの確度向上が期待される．本節では，提出手役評価関数を用いてプレイアウトを行うことで，モンテカルロ法プレイヤがどの程度強化されるのかを評価する．

対戦の組み合わせは，提案手法によるプレイヤ 1 つに対してプレイヤ MC を 4 つとする．提案手法およびモンテカルロ法のプレイアウト回数はいずれも 1500 とした．10000 ゲームを行ったときの提案手法によるプレイヤの相対得点を，表 5 に示す．提案手法によるプレイヤの相対得点は +2966〜+4447 となり，いずれの学習データを用いた場合でも提出手役評価関数を用いることでモンテカルロ法プレイヤが強化された．

3 層ニューラルネットワークと平均化パーセプトロンの学習アルゴリズムについて比較すると，相対得点の平均は 3 層ニューラルネットワークによる学習の方が少し大きい（+279）．しかし，その差は小さく，またプレイヤ kishimen や MC の札

譜データの場合には平均化パーセプトロンの方がより多い相対得点となっているため，3 層ニューラルネットワークによる学習の方が適しているという結論は導けない．

学習に用いた札譜データのプレイヤと相対得点について見ると，表 4 に示した提出手役の一致率と相対得点の間には，正の相関は見られない．一方，表 1 に示した学習に使った札譜のプレイヤの強さと相対得点の間には，一部例外はあるものの，強いプレイヤの札譜で学習する方が得られた得点が多いという結果が見られる．

6.2 学習に用いた札譜のプレイヤとの対戦による評価

提案手法におけるプレイアウトでは，学習した札譜データのプレイヤと同じような着手を行って手役の評価値を計算する．したがって，学習した札譜データのプレイヤと実際の対戦相手が同じである場合には，提案手法プレイヤの得点が向上すると予想した．

そこで，対戦相手をモンテカルロ法プレイヤ以外の場合についても対戦を行い，得られる相対得点について調査を行った．対戦の組み合わせは，提案手法によるプレイヤ 1 つに対して，プレイヤ paoonR2，kishimen，または Greedy のそれぞれを 4 つとした．また，比較対象として，ベースとなるプレイヤ MC を 1 つに対して，プレイヤ paoonR2，kishimen，または Greedy のそれぞれを 4 つとした対戦も行った．対戦は同様に 10000 ゲームで行った．対戦相手が paoonR2 である場合の相対得点およびベースのときの得点からの差分を表 6 に示す．同様に，対戦相手が kishimen である場合の結果を表 7 に，対戦相手が Greedy である場合の結果を表 8 に示す．

学習に用いた札譜データのプレイヤと実際に対戦する相手プレイヤが一致した場合に最も高い得点を得ることを期待したが，その予想に沿う結果となったのは，プレイヤ paoonR2 と Greedy の札譜を 3 層ニューラルネットワークで学習した場合のみであった．一方，プレイヤ paoonR2，すなわち，強いプレイヤの札譜を学習した場合を見ると，

表 6 プレイヤ paoonR2 との対戦結果

	相対得点 (ベースとの差)	
学習データ	NN	AP
paoonR2	−6030 (+2326)	−6412 (+1944)
kishimen	−7050 (+1306)	−6284 (+2072)
MC	−6698 (+1658)	−6326 (+2030)
Greedy	−6316 (+2040)	−6833 (+1523)
ベース	−8356	

表 7 プレイヤ kishimen との対戦結果

	相対得点 (ベースとの差)	
学習データ	NN	AP
paoonR2	+653 (+3793)	+132 (+3272)
kishimen	−153 (+2987)	−282 (+2858)
MC	+704 (+3844)	+135 (+3275)
Greedy	−96 (+3236)	−24 (+3116)
ベース	−3140	

表 8 プレイヤ Greedy との対戦結果

	相対得点 (ベースとの差)	
学習データ	NN	AP
paoonR2	+12278 (+904)	+12130 (+756)
kishimen	+11589 (+215)	+11802 (+428)
MC	+11494 (+120)	+11910 (+536)
Greedy	+12403 (+1029)	+11631 (+257)
ベース	+11374	

3 層ニューラルネットワークで学習した場合には対戦相手が paoonR2，kishimen，MC の場合に最も良い結果を得ており，また平均化パーセプトロンで学習した場合にも kishimen との対戦で良い結果を得ている．

学習に用いた札譜データのプレイヤとの対戦で良い結果が得られなかった理由についてはまだ明らかになっていないが，表 4 に示した提出手役の一致率が不十分であること，プレイアウトにおいて評価値が最大となる手を確定的に選択するのではないことが理由として考えられる．

また，第 5.1 節において，実際のゲームに参加しているプレイヤの札譜からのオンライン学習は可能であると考察した．しかし今回の結果からは，実際に対戦中の相手の札譜をオンラインで学習するよりも，強いプレイヤの札譜をオフラインで学習する方がプレイヤの強化に有利である可能性が示唆される．

7. 関連研究

7.1 棋譜からの学習

着手選択に用いる評価関数を棋譜を用いた学習によって得るという考え方は，古くは 1950 年代にチェッカーを対象とした研究において提案されている [5]．オセロでは，駒に優劣の差はないものの，どこに駒があるかは勝敗に大きく影響するため，駒の位置に対する評価値を学習で得る手法が有効である [2]．

また，将棋においても，棋譜による評価関数の学習が有効である．将棋では，駒によってその強さが異なるため，駒の強さや駒同士の関係を表すパラメータを用いた評価関数が古くから用いられていた．そのパラメータを棋譜からの学習によって調整する考え方は金子らによって提案され [7]，現在では保木が提案した手法 [15] が広く利用されている．それにより，駒の強さや位置，複数の駒の関係などの 10000 を超えるパラメータを，棋譜で指された手をもとに学習することが可能となっており，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．

一方，多人数不完全情報ゲームにおいても，棋譜からの学習が用いられている例がある．麻雀において，北川ら [8] は，その重みを棋譜（牌譜）から学習した 3 層ニューラルネットワークによって出すべき手を決める評価関数を作成した．評価関数と棋譜との一致率は，ツモ局面でおよそ 56%，鳴き局面でおよそ 89%であったが，その評価関数を用いたプレイヤのレーティングは 1318 と弱いものであった．水上ら [17] は，平均化パーセプトロン学習による 1 人麻雀プレイヤに「降り」と「鳴き」の機能を SVM 学習によって導入することにより，レーティング 1651 と平均的な人間プレイヤを上回るプレイヤの作成に成功している．

7.2 コンピュータ大貧民

2006 年からコンピュータ大貧民大会 (UECda) が開催され，大貧民に関する研究も活発に行われるようになってきた．2006 年はアプリオリなルールベース，2007 年と 2008 年は必勝手優先の探索を用いたプレイヤプログラムが優勝し，2009 年には単純なモンテカルロ法を用いたプレイヤプログラムが優勝している．2010 年以降は，モンテカルロ法に必勝手順探索，手札推定 [10]，差分学習の応用 [9] などの工夫を取り入れたプログラムが優勝している．このように，コンピュータ大貧民プレイヤにおいてモンテカルロ法は非常に重要な要素となっている．

大貧民は不完全情報ゲームであり，相手プレイヤの持つ手札は分からない．そこで，相手プレイヤの持つ手札を推定する手法およびその効果についての研究がある．2010 年の UECda において優勝したプレイヤプログラム snowl では，相手手札を推定する手法を取り入れている [10]．また，西野ら [14] は，snowl を題材に相手手札の推定をモデル化している．手札推定が当たる確率とプレイヤの強さの関係については，吉原ら [18] によって調べられている．また地曳ら [11] は，相手手札のうち最も強いカードと最も弱いカードに着目して，それらの推測がモンテカルロ法やモンテカルロ木探索に与える影響について調査している．

本研究で行った札譜データからの他プレイヤの手の学習は，大貧民においてもすでに取り組まれている．伊藤ら [6] は，対戦中に相手の出す手札を学習することを目標として，ナイーブベイズを用いた学習を提案している．提出手役の一致率は，2010 年 UECda 優勝プログラム snowl に対しておよそ 4 割であったと報告されている．また，本研究の先行研究である地曳ら [12] の研究では，3 層ニューラルネットワークを用いて，単純なモンテカルロ法プレイヤに対して提出手役の一致率は 7 割であった．地曳らの研究では 110 要素からなる入力層はであったが，本研究では 182 要素からなる入力層を用いていることが異なる．

8. まとめと今後の課題

本論文では，大貧民の札譜データから機械学習の手法を用いて作成した提出手役評価関数を用いて，プレイアウトの確度を向上することとモンテカルロ法プレイヤを強化することを提案した．入力層の評価項目を見直すこととより多くの教師デー

タを用いることで，既存研究に比べて提出手役の一致率を高めることができた．また，それらの評価関数をプレイアウトに適用したモンテカルロ法プレイヤは，単純モンテカルロ法プレイヤより強くなった．評価実験の結果から，提出手役評価関数の一致率よりも，学習データに用いた札譜のプレイヤが強いかどうかが，モンテカルロ法プレイヤの強化に強く影響することが示された．

本研究では，単純なモンテカルロ法プレイヤを強化することはできたが，既存の強いプレイヤに勝つほどまでは強くはなっていない．特に，paoonR2によって学習した提出手役評価関数は，モンテカルロ法の強化に有効であったが，提出手役一致率は67%〜71%とまだ低い．機械学習の手法の入力層の拡充により，強いプレイヤに対する一致率を高めることで，さらなる強化ができる可能性がある．また，提出手役評価関数を計算することでプレイアウト回数が少なくなることも問題である．今回の実験では，単純モンテカルロと同じプレイアウト回数での実験としたが，制限時間のある場合にはどのような条件でこの評価関数を用いるべきかを検討することも必要である．

謝辞 本研究の実験を行うにあたり，高知工科大学のIACPクラスタを利用させていただきました．

参考文献

[1] 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).

[2] M. Buro. Improving Heuristic Mini-max Search by Supervised Learning. *Artificial Intelligence*, Vol. 134, No. 1–2, pp. 85–99 (2002).

[3] Y. Freund, R.E. Schapire. Large Margin Classification Using the Perceptron Algorithm. *Machine Learning*, Vol. 37, No. 3, pp. 277–296 (1999).

[4] L. Kocsis and C. Szepesvári. Bandit Based Monte-Carlo Planning, *17th European Conference on Machine Learning (ECML 2006)*, Lecture Notes in Computer Science 4212, pp. 282–293 (2006).

[5] A.L. Samuel. Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, Vol. 3, No. 3, pp. 211–229 (1959).

[6] 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).

[7] 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).

[8] 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).

[9] 小沼 啓, 本多 武尊, 保木 邦仁, 西野 哲朗. コンピュータ大貧民に対する差分学習法の応用. **研究報告ゲーム情報学（*GI*）**, Vol. 2012-GI-27, No. 1, pp. 1–4 (2012).

[10] 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).

[11] 地曳 隆将, 松崎 公紀. 大貧民において不完全情報性がモンテカルロ法によるプレイヤに与える影響の調査. **情報処理学会研究報告**. *GI, [ゲーム情報学]*, Vol. 2012-GI-28, No. 6, pp. 1–8 (2012).

[12] 地曳 隆将, 松崎 公紀, 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. **情報処理学会研究報告**, Vol. 2014-GI-31, No. 15 (2014).

[13] 電気通信大学. UEC コンピュータ大貧民大会, `http://uecda.nishino-lab.jp/2014/` (2014).

[14] 西野 順二, 西野 哲朗. 大貧民における相手手札推定. **研究報告数理モデル化と問題解決** *(MPS)*, Vol. 2011-MPS-85, No. 9, pp. 1–6 (2011).

[15] 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).

[16] 松原 仁 (編), 美添 一樹, 山下 宏 (著). **コンピュータ囲碁—モンテカルロ法の理論と実践**. 共立出版 (2012).

[17] 水上 直紀, 中張 遼太郎, 浦 晃, 三輪 誠, 鶴岡 慶雅, 近山 隆. 多人数性を分割した教師付き学習による 4 人麻雀プログラムの実現, **情報処理学会論文誌**, Vol. 55, No. 11, pp. 2410–2420 (2012).

[18] 吉原大夢, 大久保誠也. コンピュータ大貧民における手札推定の有効性について. **情報処理学会研究報告**, Vol. 2013-GI-30, No. 4 (2013).