# 方策勾配を用いた教師有り学習による コンピュータ大貧民の方策関数の学習と モンテカルロシミュレーションへの利用

大渡 勝己[1,a)]　田中 哲朗[2,b)]

**概要**：大貧民は多人数不完全情報ゲームに分類され，日本中で広く親しまれているゲームである．大貧民のコンピュータプログラムの大会（UECda）も行われており，近年ではモンテカルロ法を用いたプログラムが上位を占めている．本研究では，大貧民の知識を用いた方策関数を設計し，そのパラメータを方策勾配を用いた教師有り学習によって過去のプログラムの棋譜から学習した．その結果，公開されている過去のコンピュータ大貧民のプログラムと比較し，モンテカルロ法を使わないプログラム，使うプログラムのいずれとしても過去最高レベルの強さを達成することが出来た．さらに，他プログラムの棋譜を用いずとも，プログラムの自己対戦棋譜からの学習を繰り返すことで，同等の強さまでプログラムを強くすることに成功した．

**キーワード**：大貧民，大富豪，モンテカルロシミュレーション，教師有り学習，方策勾配

# Supervised learning of policy function based on policy gradients and application to Monte Carlo simulation in Daihinmin

Katsuki Ohto[1,a)]　Tetsuro Tanaka[2,b)]

## 1. はじめに

近年では人工知能は金融サービスなど様々な分野での応用が進んでおり，現実世界での意思決定へのコンピュータの介入が今後さらに増えることが予想される．

一方，大貧民は日本において最も人気のあるトランプゲームのひとつである [1]．相手の手札が見えないことや，ランダムなカード配布による運の要素が強いこと，前の試合の結果によって次の試合の有利不利が変わることなど，現実世界を模していると言われることもある．

そのため，コンピュータに大貧民をプレイさせる取り組みは，今後コンピュータが現実社会での意思決定に深く関わる上での試金石となり得る．

毎年，電気通信大学にてコンピュータ大貧民大会が開催され，近年ではモンテカルロシミュレーション（モンテカルロ法）を用いたプログラムが良い成績を収めている．

大貧民に対して機械学習を適用する研究も行われており，須藤ら [2] や飯田ら [3] はシミュレーションバランシングによって，モンテカルロシミュレーション中の方策関数を学習した．

また，岡ら [4] はニューラルネットよる方策関数にて，棋譜の手が選ばれやすくなるように学習を行った．その結果過去のプログラムに対して高い一致率を達成し，学習した方策関数をシミュレーションに使うことでプログラムの強化に成功した．

一方で，機械学習主体で作られたプログラムのうち，2015年時点でのコンピュータ大貧民大会の上位プログラムに匹

[1] 東京大学大学院総合文化研究科
Graduate School of Arts and Sciences, The University of Tokyo
[2] 東京大学情報基盤センター
Information Technology Center, The University of Tokyo
[a)] ohto@tanaka.ecc.u-tokyo.ac.jp
[b)] ktanaka@tanaka.ecc.u-tokyo.ac.jp

敵する強さのものは報告されていない．

実際に 2012 年〜2014 年までの無差別級（機械学習手法を用いた部門）の優勝プログラムは，機械学習部分以外の人手による成果も大きい．そのため，これらのプログラムの強さを再現することは難しい．

そこで本研究では，先行研究とは異なる機械学習手法にてプログラムの強化を試みた．また，先行研究においては機械学習によって学習した方策関数そのままでのプレーの強さについて言及が少なかったため，方策関数そのままでの対戦実験も行った．

本稿における提案は以下の 2 点である．

- 必勝探索や手札推定を加えた，大貧民プログラムのアルゴリズム
- 方策関数の学習手法の大貧民への適用

それぞれ 3 節と 5 節で概要を説明する．

## 2. コンピュータ大貧民大会

UEC コンピュータ大貧民大会（UECda）[5] は 2015 年までに 10 回開催されている．大貧民のルールには様々なバリエーションがあるが，本研究はこの大会のレギュレーションに沿っている．

### 2.1 UEC 標準ルール

UEC コンピュータ大貧民大会で扱われている UEC 標準ルールについて説明する．ただし，大貧民全般で共通しているルールや，本稿の内容に直接関係しないものは省いている．詳細なルールについては大会の公式ホームページ [5] を参照されたい．

なお，以下で用いた用語はあくまで本稿における呼び名であり，必ずしも大貧民における一般的な用語とは限らない．

- 参加人数は 5 人．
- ジョーカー 1 枚を含む 53 枚のトランプ（カード）を用いる．
- カードは特定の組み合わせ（役）として場に出すことができる．
- 役にはカード 1 枚（単体），同じ数字のカード複数（グループ），同じスートの連番（階段）がある．
- カードの数字によって強さが決まっている．通常時には 3 が最も弱く，2 が最も強い．
- 場に役が出ている場合には，同じ種類（グループ，階段は枚数も同じ）のより強い役だけを出すことができる．
- 手持ちのカードが無くなった（上がり）順に順位を付け，最後に一人が残るまで続ける．
- まだ上がっていない全員がパスをした場合と，以降の特別な場合に，場から役が無くなる（場が流れる，と呼ぶ）．このとき，最後に役を出したプレーヤー（場役主）の手番になる（場を取る，と呼ぶ）．
- 流れる基準が「全員のパス」であるため，流れずに同じプレーヤーが連続して役を出すことができる．
- 4 枚以上のグループと 5 枚以上の階段が場に出ると数字の強さの順序が逆転する（革命）．
- 場に出ている役と同じスート構成の役が出ると，以降場が流れるまで同じスート構成の役しか出せなくなる（スートしばり）．
- 単体のジョーカーは非革命時・革命時共に最強の単体役として出せる．
- 8 のカードを含む役を出すと場が流れる（8 切り）．
- ジョーカーは他のカード 1 枚の代わりとしてグループ・階段に含めることが出来る．このとき革命，スートしばり，8 切りのルールも同じように適用される．
- 単体のジョーカーが場に出ている際に，単体のスペードの 3 を出すことが出来て，場が流れる（スペ 3 返し）．
- 上がり順によって次の試合の階級が決定する．1 位から 5 位まで順に大富豪，富豪，平民，貧民，大貧民という階級になる．
- カード配布後，階級に応じたカードの交換を行う．大富豪と大貧民の間で 2 枚，富豪と貧民の間で 1 枚のカードを交換する．下位（貧民，大貧民）は手札内の最も強いカードを渡し，上位（大富豪，富豪）はそれを受け取った後にどのカードを渡すかを選ぶことができる．ただし，上位はサーバーからのカード配布時点で下位から受け取ったカードをすでに持っており，どのカードを渡されたか知ることができない．
- ダイヤの 3 を持つプレーヤーが最初に手番を持つ．
- 各ゲーム後に上がり順に 5 点，4 点，3 点，2 点，1 点が与えられ，大会では合計得点で勝敗を決める．
- 最初のゲームでは全員が平民である．その後は一定試合数（本研究では 100 試合）ごとに階級が初期化される．全員が平民の場合カード交換は行われない．

### 2.2 部門

2012 年〜2015 年の大会では，2 つの部門に分かれて大会が行われた．

- ライト級... 実行時間が短く，機械学習手法を用いないプログラムの部門
- 無差別級... ライト級の条件に当てはまらない，主に機械学習手法を用いたプログラムの部門

無差別級においては，長らくモンテカルロ法を用いたプログラムが優勝を続けている．モンテカルロ法を用いたプログラムと，それ以外のプログラムの強さには大きな開きがあるのが現状である．

## 3. 提案アルゴリズム

### 3.1 行動決定方法の概略

提案アルゴリズムにおいては，まず必勝が確定する交換カードや役の探索を行う．この際，残り 2 人になって相手の手札が確定した場合には相手の着手も含めた探索を行うが，それ以外の場合は計算量の削減のため，相手に 1 枚も出させずに上がるなどの簡単な必勝かどうかだけを探索している．

必勝の行動が発見できなかった場合の行動決定は，本研究では「方策関数のみ」「モンテカルロ法を利用」の 2 通りで実験を行った．それぞれの行動決定方法は以下の通りである．

- 方策関数のみの場合（以下 POLICY と呼ぶ）
  ... 方策関数にて最大の確率がついた行動を選択
- モンテカルロ法を用いる場合（以下 MC と呼ぶ）
  ... モンテカルロシミュレーションを行って末端評価点（後述）の期待値が最も高い行動を選択

### 3.2 手札推定

大貧民は不完全情報ゲームであり，相手の持つ手札を知ることが出来ない．そのためシミュレーションを行う際には，相手の状態を推定し，実際に手札を配置する必要がある．

手札推定に関してはこれまでいくつかの報告がなされている．西野ら [6] は 2011 年当時のプログラムにおいて，手札推定の効果は小さいと報告した．一方で，吉原ら [7]，平嶋ら [8] は手札推定の有効性に言及している．

提案アルゴリズムによるプログラムにおいては，事前実験により，相手手札をランダムに分配するよりも，以下のアルゴリズムによる推定を加えたほうが成績が良かった．そのためモンテカルロ法での行動決定の際には手札推定を必ず行っている．

手札推定は以下の手順で行っている．

(1) カード交換後（カード交換が無いゲームでは，手札が配られた直後）のカード配置に近い分布に従う手札配置を一定個数生成
(2) それぞれの配置から，当試合の試合進行が生まれる尤度を計算
(3) 尤度最大のものだけを採用し，モンテカルロシミュレーションに使用する手札配置セットに加える

使用する手札配置が一定個数に達するまで（1）～（3）を繰り返すことで手札配置サンプルを生成する．

（1）～（3）のうち，(2) では，相手の着手方策が，提案手法によって学習した方策に従っているという仮定をおいて尤度を計算する．

ただし，役提出の尤度計算においては，方策関数の計算の前に必勝着手の探索を行っている．

本手法における方策モデルは，必勝着手を必ず選ぶことを前提としているため，必勝着手が選ばれなかった場合には尤度 0 とするのがモデルに忠実ではある．

しかし，対戦相手が自分と同様の必勝探索を行うとは限らないため，そのような手札配置が有り得ないとは言い切れない．さらに，尤度 0 の手札配置が頻発すると手札推定が機能しなくなる可能性がある．そのため，本手法では必勝逃しの尤度を 0 にする代わりに，方策関数の計算の際に必勝着手に大きなボーナスを与えている．このボーナスの値は本研究にて学習の対象には含めていない．

（1）では，**表 1** に示す手順によって，手札配置を生成している．なお表の説明において，「配る」はランダムなカード配布を意味し，「交換」は学習したカード交換方策によって行う．

基本的には以上の手順によってカード交換後の手札配置を生成したが，この方法では各プレーヤーが多くのカードを使用した中終盤には計算量が組み合わせ爆発を起こす．

そのため，一つの手札配置を作るまでのループに回数制限を設け，回数制限を超えた場合は失敗とした．失敗が一定回数あった場合には，それぞれのカードがどの階級のプレーヤーに所持されているかの割合を事前計算したテーブルによってランダムにカードを配る簡単な手法に切り替えた．

本手法で用いた手札配置の作成アルゴリズムは計算量が大きいため，作成個数を制限（本研究では最大 128 個）して同じ手札配置を何度も使い回している．

### 3.3 モンテカルロシミュレーション

MC におけるシミュレーションにおいては，まず使用する手札配置サンプルを選ぶ．

その後は方策関数（後述）によって一手ずつシミュレーションを進める．

ただし例外として，残り 2 人になった場合には探索してどちらが勝つかを調べ，シミュレーションは終了する．また，3 人以上残っている場合でも，簡単な必勝着手の探索を行って必勝着手が見つかった場合には，方策関数を計算することなくその手でシミュレーションを進める．

なお，本研究では自分の順位以外は考慮していないため，シミュレーション中に自分が上がった場合には，そこでシミュレーションを打ち切っている．

大貧民においては，カード交換というルールにより 1 試合の順位が次の試合にも影響する．本研究では，過去のプログラムの試合における階級遷移確率から，階級が初期化されるまでに獲得する合計得点の期待値を計算し，その値を元にシミュレーションの末端評価値を定めた．

表 1　相手手札配置の作成方法

| |
|---|
| 自分が大富豪・富豪の場合 |
| a. 配布後，献上が終わった後のカードの情報から<br>交換相手の「未使用カードの所持確率分布」の表を作り，<br>表を用いて交換相手に現在の手札枚数分を重み付け配布<br>b. 残りカードを，「自分以外の交換ペア」と「平民」に配る<br>c. 自分以外の交換ペアの手札と両者の使用カードを混ぜ<br>配布直後の枚数で両者に配る<br>d. 自分以外の交換ペアの間でカード交換を行う<br>e. 自分以外の交換ペアが既に使用した<br>カードとの間に矛盾があれば，a に戻る<br>この配置で決定 |
| 自分が平民の場合 |
| a. 未使用カードを「大富豪-大貧民ペア」と<br>「富豪-貧民ペア」に配る<br>b. 大富豪-大貧民ペアの手札と両者の使用したカードを混ぜ<br>大富豪と大貧民に配る<br>c. 大富豪と大貧民の間でカード交換を行う<br>d. 大富豪と大貧民に配られた手札と，<br>両者が既に使用したカードとの間に矛盾があれば，a に戻る<br>e. 富豪-貧民ペアの手札と両者の使用カードを混ぜ<br>富豪と貧民に配る<br>f. 富豪と貧民の間でカード交換を行う<br>g. 富豪と貧民に配られた手札と，<br>両者が既に使用したカードとの間に矛盾があれば，a に戻る<br>この配置で決定 |
| 自分が貧民・大貧民の場合 |
| a. 未使用カードを「自分の交換相手」と「それ以外」に配る<br>b. 自分が交換相手からもらった札を返し，<br>交換相手から自分へのカード交換を行う<br>c. 実際に自分がカード交換で手に入れた札と違ったら，a に戻る<br>d. 残りカードを「自分以外の交換ペア」と「平民」に配る<br>e. 自分以外の交換ペアの手札と両者の使用カードを混ぜ<br>配布直後の枚数で両者に配る<br>f. 自分以外の交換ペアの間でカード交換を行う<br>g. 両者が既に使用したカードとの間に矛盾があれば，a に戻る<br>この配置で決定 |
| カード交換が無いゲームの場合 |
| a. 未使用カードを，現在の手札枚数分配る<br>この配置で決定 |

### 3.4　シミュレーションの割り振り

須藤ら [2] に倣い，どの行動候補にシミュレーションを割り振るかの決定を UCB1-tuned[9] にて行っている．

ただし，評価が高い候補への集中を防ぐため，本来の UCB1-tuned の式からバイアス項に係数$\sqrt{6.0}$ を掛けている．また各行動候補の最低シミュレーション回数を設ける，行動の候補が 2 つの場合には同じ回数のシミュレーションを行うといった調整も加えている．

## 4.　方策関数

カード交換，役の提出のための方策関数はいずれも，須藤ら [2] と同じく，線形関数による softmax 方策とした．

以下では，状態 $s$ で行動 $a$ を取る際の特徴ベクトルを $\phi(s,a)$，各特徴に対する重みベクトルを $\theta$ と表す．このとき状態 $s$ で可能な行動の集合を $A$ としたときに，行動 $a \in A$ を 取る確率を

$$\pi_\theta(s,a) = \frac{e^{\phi(s,a)\cdot\theta/T}}{\sum_{b\in A} e^{\phi(s,b)\cdot\theta/T}} \tag{1}$$

と定める．ただし $T$ は温度パラメータと呼ばれ，この値を大きくすると方策はランダムに近づく．本研究では方策関数でプレーする場合には $T=0$，手札推定とシミュレーションでは $T=1$ としている．

特徴ベクトルの各要素について，本研究で用いたものを**表 2**，**表 3** にまとめた．

評価要素の作成においては，先行研究にて有効性が確認された要素を数多く取り入れた．

「手札の構成要素」は須藤ら [2] の用いた評価要素を場合によって分けたものである．

また，「スートしばりをかける際に，同じスートでの最強カードを自分が持つかどうかで場合分け」「同じ枚数の数字の組数」は田頭ら [10] が有効性を報告している．

さらに，「強い札と 8 はグループを崩して出す」戦略を坂田ら [11] が強化学習の結果として獲得したと述べている．

「手札の最小分割数」は上がりまでの最短手数を意味しており，多くのプログラムが類似のアイデアを用いている．

ただし独自の実装のため，先行研究と必ずしも同一の特徴を見ているとは限らない．また評価要素説明の字義とは微妙に異なった実装になっているものもある．

表 2　カード交換方策の評価要素

| | |
|---|---|
| 交換後の手札の評価 | |
| 残り手札の構成要素 | |
| （須藤ら [2] の「snowl」の評価要素のうち | |
| 非革命時のみ，かつジョーカーの項は除く） | 82 |
| ダイヤ 3 を持っている | 1 |
| ジョーカーとスペード 3 の配置 | |
| （自分が両方，自分と自分以外が片方ずつ） | 3 |
| 非革命時に一番強い数字，二番目に強い数字 | 2 |
| 非革命時に一番弱い数字，二番目に弱い数字 | 2 |
| 渡す札の情報 | |
| 同じ数字のカード 2 枚を渡す | 1 |
| 同じスートの連続した数字のカード 2 枚を渡す | 1 |
| 同じスートの 1 つ飛ばしの数字のカード 2 枚を渡す | 1 |
| 計 | 94 |

## 5.　学習手法

方策関数の学習は，方策勾配を用いた教師有り学習 [12] によって行った．この手法は教師の手が選ばれやすくなるように方策のパラメータを調整する手法である．これまでのところ，大貧民において有効という報告はない．

方策勾配を用いた教師有り学習における誤差関数は，教

表 3　役の提出方策の評価要素

| | |
|---|---|
| 着手後の手札の評価 | |
| 残り手札の構成要素 | |
| （須藤ら [2] の「snowl」の評価要素を | |
| 空場，パスで場がとれるとき，それ以外で場合分け） | 498 |
| ジョーカー，スペード 3 が残っている | |
| （自分が両方，自分と自分以外が片方ずつ） | 3 |
| $\ln(\frac{\text{着手後手札の平均強さ}}{\text{着手前手札の平均強さ}}) \times$ 着手前残り手札枚数 | 1 |
| 残り手札の最小分割数の増加量（空場，空場でない） | 2 |
| パス以外の着手の評価 | |
| 空場の時，着手の枚数 | 4 |
| 階段でないとき同じ枚数の数字の組の数 | |
| （単体ジョーカー，ジョーカー使用時，不使用時） | 3 |
| スートしばりをかけるとき，同じスートでの | |
| 最高ランクの札を自分が持っているか | |
| （持っている，持っていない，複数枚でのしばり） | 3 |
| 非革命時からの革命 | |
| （残りプレーヤー内での相対的階級ごと） | 5 |
| スペ 3 返し（パスでも場が取れるか） | 2 |
| スペード 3 を誰かが持っている場合の単体ジョーカー | |
| （自分で返せる，返されない，返される可能性あり） | 3 |
| 階段である（空場，空場でない） | 2 |
| 階段のとき，出す前の自分の手札枚数（空場のみ） | 1 |
| 空場でないときに 8 切りまたは | |
| 場が取れる数字のグループを崩して出す | 3 |
| パスでも場が取れるときに | |
| 次の空場で確実に場が取れる着手 | 1 |
| 8 より弱い組み合わせが | |
| 複数あるときに空場で階段でない 8 を出す | 1 |
| 8 切りを出すとき，着手後手札枚数とその 2 乗 | 2 |
| 自分が持っている中で最も弱いカードを出す | |
| （革命時，非革命時） | 2 |
| パスの評価 | |
| 全体の残り手札枚数 3 段階 | 3 |
| パスをしても場が取れるか | 2 |
| 現在場役主が場を取った場合に | |
| 自分が何人目か | 4 |
| パスをした後の残り人数と | |
| 場役主のカード枚数組み合わせ | 32 |
| 計 | 577 |

師の確率的方策を $\pi_*$ とするとき

$$L(\pi_*, \pi_\theta) = \sum_{b \in A} \pi_*(s,b) \ln \frac{\pi_*(s,b)}{\pi_\theta(s,b)} \tag{2}$$

という式によって与えられる．特に教師の方策 $\pi_*$ が決定的な場合には，$A$ における教師の行動を $x$ として

$$L(\pi_*, \pi_\theta) = -\ln \pi_\theta(s,x) \tag{3}$$

という式になる．

大貧民における理想的な方策や，教師とするプレイヤの方策は必ずしも決定的とは言えない．一方で多人数不完全情報ゲームである大貧民において，全てにおいて「同一」の局面は滅多にない．

本研究では棋譜からの学習を行うため，教師の着手方策は決定的と仮定し，誤差関数は式 3 とした．

方策関数が式 1，誤差関数が式 3 のとき，最急降下法による重みベクトル $\theta$ の更新を以下の式 4 で行った．ただし学習率を $\alpha$ とする．

$$\theta \leftarrow \theta + \frac{\alpha}{T}[\phi(s,x) - \sum_{b \in A} \pi_\theta(s,b)\phi(s,b)] \tag{4}$$

提案アルゴリズムの行動決定において，手札推定の際以外に必勝の局面で方策関数を呼ぶことはない．また残り 2 人で負け局面の場合の棋譜の着手は，教師として適当ではない可能性がある．

以上により，学習に用いる局面は，3 人以上が残っており，必勝手探索により必勝手が見つからなかった局面のみとした．

## 6. プログラムの強さの評価方法

多人数ゲームにおいて，どのような集団内での勝率を重視するべきかは時と場合によって異なるため，強さを測る統一的な方法はない．

本研究においては，学習を行ったプログラムの強さの評価指標には，2015 年のコンピュータ大貧民大会決勝のプログラムに対する 1 試合あたりの平均得点を用いた．

具体的には，POLICY の対戦相手はライト級決勝に残ったプログラムとし，MC の対戦相手は無差別級決勝に残ったプログラムとした．

対戦相手のプログラム一覧を**表 4** にまとめた．無差別級は上位 4 プログラム，ライト級は 2 位のプログラムが公開されていなかったため，1，3，4，5 位のプログラムとした．

POLICY は残り 2 人時の全探索などを行うが、それでも計算量は対戦相手プログラムと同程度であり，2015 年大会ライト級の出場要件を満たしている．

一方 MC においては，2015 年のコンピュータ大貧民大会無差別級の出場要件を満たすため，シミュレーションの回数はカード交換，役の提出ともに 5000 回で固定した．この場合の計算時間は，2015 年のコンピュータ大貧民大会の本番環境にて，8 スレッドで動かせば無差別級の計算量制限を通過可能な程度であった．

POLICY の対戦相手には 2015 年大会ライト級優勝の kou2 が含まれており，MC の対戦相手には 2015 年大会無差別級優勝の wisteria が含まれている．これまでのところ，モンテカルロ法を使わないプログラムと使うプログラムのそれぞれで，この 2 つより強いプログラムの報告は無い．そのため，これらのプログラムとの平均得点を比較することで，提案手法によるプログラムが過去最高レベルを超えているかどうかを判断した．

## 7. 実験

提案手法によるプログラムの強さの検証のため，2 種類

表 4　対戦実験の相手プログラム

| 方策関数プレイヤの対戦相手 |
|---|
| kou2, KT, KZ2015, masa |
| モンテカルロ法プレイヤの対戦相手 |
| wisteria, halfsnooze, yudai, jn15 |

表 5　教師として用いたプログラム

| |
|---|
| default, jnishino, Nakanaka, |
| Party, Party2, kou+, kou2, |
| fumiya, snowl, crow, paoon, |
| beersong, FujiGokoro, wisteria, yudai |

の実験を行った．

1 つ目の実験では，過去のプログラムの棋譜を利用して方策関数を学習し，プログラムの強さを評価した．以下ではこれを実験 1 と呼ぶ．

2 つ目の実験では，過去のプログラムの棋譜を用いず，自分の棋譜だけを学習に使用した．学習した方策関数を用いてモンテカルロシミュレーションを行い，さらに棋譜を作って学習に使うことを繰り返した．以下ではこれを実験 2 と呼ぶ．

### 7.1　実験 1 手法

**表 5** にある全 15 の教師プログラムの自己対戦棋譜をそれぞれ 50000 試合作成し，40000 試合を学習データ，残り 10000 試合をテストデータとした．

役提出の学習は反復回数 50 回，カード交換の学習は，カード交換が 1 試合に 2 回（大富豪から大貧民，富豪から貧民）と少ないため反復回数 150 回とした．それ以外のパラメータは，以下の通りとした．

- 学習率 $\alpha$... 0.00005 / 特徴要素の値の分散
- 温度 $T$...1
- L1 正則化係数...0
- L2 正則化係数...0.0000001

学習はそれぞれの教師プログラムに対して別々に行い，15 通りの重みパラメータ（式 1 における $\theta$）を得た．

次に，それぞれのパラメータを読み込んだ POLICY，MC の強さを検討するため，表 4 に示した相手との試合を POLICY は 100000 試合，MC は 25000 試合行った．

また，学習のための棋譜作成に用いた教師のプログラム内で，同じプログラムが複数参加しない全組み合わせで各対戦 100 試合のリーグ戦（計 330300 試合）を行った．以後，このリーグ戦における各プログラムの平均得点を，教師プログラムの平均得点として扱う．

### 7.2　実験 1 結果

全教師プログラムについて，学習の結果，方策関数にて最大の得点が付いた行動と棋譜の行動との一致率を**表 6** にまとめた．なお，合法着手が 1 つの局面，役提出において必勝着手が見つかった局面，残り 2 人の局面は，方策関数が呼ばれないため平均分岐数や一致率の計算には含めていない．

特に役提出においては教師プログラム間での平均分岐数の差が大きかったので，そのデータも合わせて載せた．

表 6　教師の対戦棋譜からの学習結果

| 教師 | 交換一致率 | 役提出一致率（平均分岐数） |
|---|---|---|
| default | 0.873 | 0.948 (5.602) |
| jnishino | 0.972 | 0.921 (6.311) |
| Nakanaka | 0.900 | 0.836 (6.602) |
| Party | 0.708 | 0.694 (6.563) |
| fumiya | 0.714 | 0.710 (5.701) |
| Party2 | 0.726 | 0.694 (6.473) |
| kou+ | 0.804 | 0.801 (6.606) |
| kou2 | 0.900 | 0.794 (6.534) |
| crow | 0.709 | 0.668 (6.347) |
| snowl | 0.715 | 0.659 (6.343) |
| yudai | 0.719 | 0.684 (6.828) |
| paoon | 0.471 | 0.624 (6.468) |
| beersong | 0.567 | 0.628 (6.358) |
| FujiGokoro | 0.980 | 0.655 (6.572) |
| wisteria | 0.611 | 0.648 (6.527) |

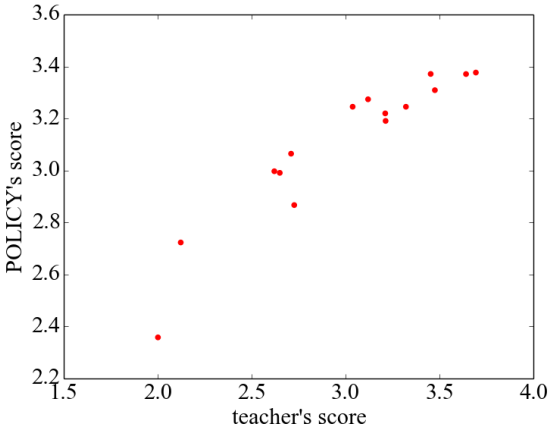


図 1　教師の平均得点と方策関数プレイヤの平均得点の関係

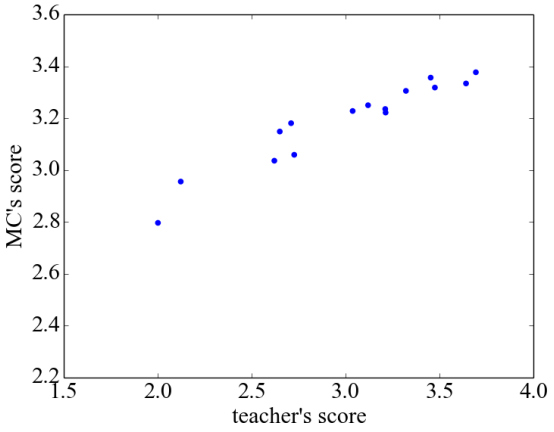


図 2　教師の平均得点とモンテカルロ法プレイヤの平均得点の関係

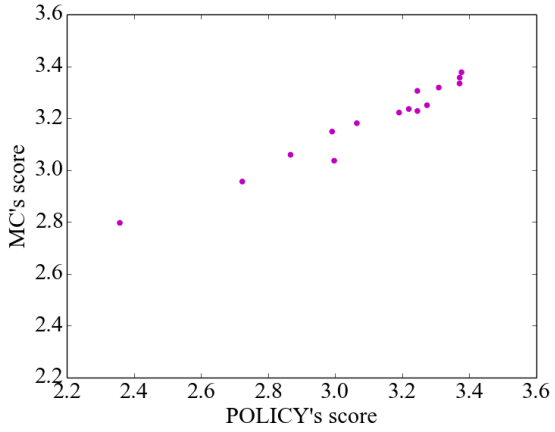


**図 3** 方策関数プレイヤの平均得点と
モンテカルロ法プレイヤの平均得点の関係

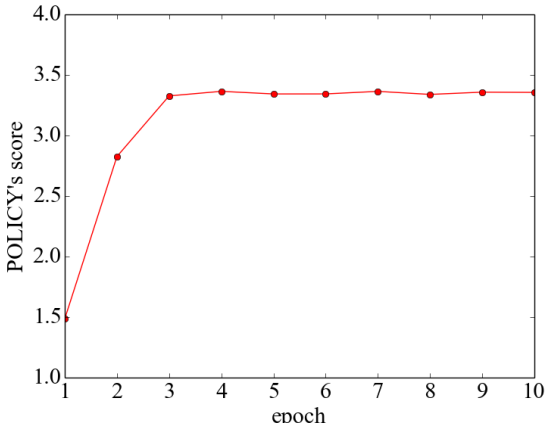


**図 4** 自分の棋譜から学習した方策関数プレイヤの
世代ごとの平均得点

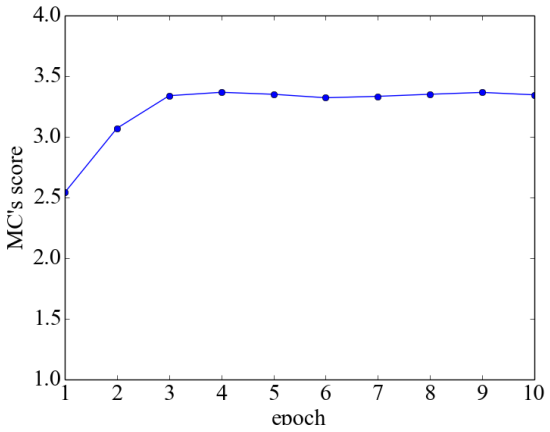


**図 5** 自分の棋譜から学習したモンテカルロ法プレイヤの
世代ごとの平均得点

一致率計算の方法が異なるので単純には比較できないが，岡ら [4] の結果（paoon に一致率 0.714）には及ばなかった．

次に，教師の平均得点と，教師から学習したパラメータを読み込んだ POLICY の平均得点を**図 1** に表した．図 1 より，教師の強さと，教師から学習した POLICY の強さに正の相関関係が見られた．

MC の平均得点についても同じように，教師の平均得点との関係を**図 2** に表した．図 2 より，教師の強さと，教師から学習した MC の強さにも正の相関関係が見られた．

さらに，**図 3** に，同じ教師の棋譜から学習した POLICY と MC の得点率の関係を示した．この結果，方策関数でのプレーの強さとモンテカルロ法での強さに，はっきりと正の相関関係が見られた．

### 7.3 実験 2 手法

実験 2 は以下の手順で行った．

1. パラメータを全て 0 に初期化する
2. 現在のパラメータによる MC 同士の対戦棋譜を作成
3. 2 で作成した棋譜から，提案手法により新たにパラメータを学習
4. 2 に戻る

パラメータ全て 0 を第 1 世代とし，学習と対戦棋譜の作成を繰り返して第 10 世代までの方策パラメータを作成した．対戦棋譜作成時のシミュレーション回数は対戦実験時と同じ 5000 回とし，学習のパラメータも全て実験 1 と同じ設定した．

それぞれの世代で，POLICY と MC 共に実験 1 と同様の対戦相手，試合数で対戦実験を行った．

### 7.4 実験 2 結果

**図 4** に POLICY の世代ごとの平均得点を示した．一方，**図 5** に MC の世代ごとの平均得点を示した．

POLICY，MC ともに開始から 3，4 世代まで得点率が上昇を続け，以降は大きな変化はなかった．

**表 7** 他プログラムからの学習と自分からの学習の
上位の結果比較

| | 他プログラムからの学習 | | 自分からの学習 | |
|---|---|---|---|---|
| | 平均得点 | 教師 | 平均得点 | 世代 |
| 方策関数プレイヤ | 3.377 | wisteria | 3.365 | 4 |
| | 3.371 | paoon | 3.364 | 7 |
| | 3.371 | FujiGokoro | 3.358 | 9 |
| モンテカルロ法プレイヤ | 3.377 | wisteria | 3.367 | 4 |
| | 3.356 | paoon | 3.366 | 9 |
| | 3.334 | FujiGokoro | 3.351 | 8 |

さらに，自分の棋譜から学習したプログラムの対戦における平均得点を，実験 1 にて他プログラムの棋譜から学習した場合の平均得点と比較したのが**表 7** である．表 7 には，実験 1 と実験 2 の対戦実験結果にて，POLICY と MC それぞれで平均得点が上位 3 位までの結果を載せた．

表 7 にて，自分の棋譜から学習したプログラムの平均得点は，過去のプログラムから学習した場合の平均を超えはしなかったが，両者の平均得点の差は 1 試合あたり 0.01 点

表 8　kou2（2015 年大会ライト級優勝）
wisteria（2015 大会無差別級優勝）との得点差上位

| | 他プログラムからの学習 | | 自分からの学習 | |
|---|---|---|---|---|
| | 得点差 | 教師 | 得点差 | 世代 |
| 方策関数 | 0.090 | wisteria | 0.078 | 7 |
| プレイヤと | 0.080 | paoon | 0.069 | 9 |
| kou2 との点差 | 0.067 | FujiGokoro | 0.066 | 4 |
| モンテカルロ法 | 0.132 | wisteria | 0.111 | 5 |
| プレイヤと | 0.079 | paoon | 0.097 | 4 |
| wisteria との点差 | 0.067 | FujiGokoro | 0.082 | 10 |

程度であった．

### 7.5　過去のプログラムとの比較

**表 8** に，POLICY と MC それぞれの対戦実験において，対戦相手に含まれる kou2，wisteria との得点差の上位 3 位までを載せた．結果，方策関数のみのプレー，モンテカルロ法でのプレーのいずれにおいても，他プログラムの棋譜からの学習，自分の棋譜からの学習で共に，対戦相手の中に含まれた過去最高のプログラムより平均得点が高くなった．

表 8 には全てを載せていないが，自分の棋譜からの学習では，第 3 世代以降全ての世代で過去のプログラムを平均得点で上回った．

## 8.　考察

過去のプログラムからの学習を行った実験 1 においては，より強いプログラムの棋譜を用いることで方策関数プレイヤ，モンテカルロ法プレイヤが共に強くなるという結果が得られた．この結果から，本研究で用いた方策関数の特徴要素と学習手法はある程度の妥当性があったと考えられる．

また，同じパラメータでの方策関数プレイヤとモンテカルロ法プレイヤの強さに強い相関が見られた．これは，モンテカルロ法プレイヤの開発においても，方策関数そのままで対戦実験を行うことの有用性を示唆している．

強さの判定に相当の試合数が必要な大貧民において，モンテカルロ法プログラムの対戦実験は開発サイクルを遅らせる最大の要因となっている．

そのため，本研究の成果は，大貧民や類似ゲームのプログラムの開発効率の向上に大きく寄与するものと考えている．

自分自身の棋譜を用いて学習を繰り返した実験 2 において，世代を重ねてプログラムが強くなり続けることはなかったが，これは学習手法以前に，方策関数の表現能力の限界が来ている可能性がある．使用する特徴要素をさらに充実させた結果，どのような結果が得られるかは興味深い．

一方で，別のプログラムの棋譜を使用した場合と同程度の強さのプログラムが出来たことは，ゲームに対する過去の蓄積を直接的に使うことなくプログラムを強化した点で大きな成果と考えている．

一般に自己対戦棋譜からの学習では，出現局面の偏りが学習の妨げとなり得る．しかし大貧民では初期手札がランダムに配られるので，自己対戦であってもある程度幅広いサンプリングが出来たために，反復的な学習が上手くいったのではないかと考えている．

この結果は，他のルールの大貧民や，似た構造を持つ他のゲームにおいても，ゲームの性質に関する知識さえあれば反復的に強いプログラムを作れることを示唆している．

一方で，本研究においては，シミュレーション中の方策は全て対称であり，対戦相手の着手傾向の違いを考慮しておらず，その点は今後の課題である．しかし本研究にて行った教師有り学習は，対戦中の相手の行動方策を直接学習することが可能である．本研究の成果も交え，相手に応じてプレーを変化させられるプレイヤの研究が進むことが期待される．

### 参考文献

[1] 西野哲朗 : 第 1 回 UEC コンピュータ大貧民大会 (UECda-2006) の実施報告, 情報処理学会誌, Vol. 48, No. 8, pp. 884 - 888 (2007).
[2] 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講演予稿集, 電気通信大学 (2010).
[3] 飯田伸也, 藤田悟 : 大貧民におけるシミュレーション・バランシングを用いた方策学習, 情報処理学会第 77 回全国大会講演論文集, No. 1, pp. 93 - 95 (2015).
[4] 岡和人, 松崎公紀 : 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, 第 56 回プログラミングシンポジウム予稿集, pp 13 - 24 (2015).
[5] 電気通信大学. UEC コンピュータ大貧民大会, http://uecda.nishino-lab.jp/ (2016. 2. 11 閲覧).
[6] 西野順二, 西野哲朗 : 大貧民における相手手札推定, 2011-MPS-85, No. 9, pp. 1 - 6 (2011).
[7] 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, 2013-GI-30, No. 4, pp. 1 - 6 (2013).
[8] 平嶋遼馬, 鈴木徹也 : コンピュータ大貧民でのモンテカルロ法における相手手札推定率と勝率との関係, 情報処理学会第 76 回全国大会講演論文集, No. 1, pp. 607 - 608 (2014).
[9] P. Auer, N. Cesa-Bianchi, and P. Fischer : Finite- time Analysis of the Multiarmed Bandit Problem. Machine Learning, Vol. 47, pp. 235 - 256 (2002).
[10] 田頭幸三, 但馬康宏, 菊井玄一郎 : 大貧民プログラムにおけるヒューリスティック戦略の評価, 情報処理学会研究報告, 2015-GI-34, No. 9, pp. 1 - 6 (2015).
[11] 坂田浩平, 大橋健 : 大富豪におけるペア温存戦略基準の獲得, ゲームプログラミングワークショップ 2008 論文集, No. 11, pp. 67 - 72 (2008).
[12] 五十嵐治一, 森岡祐一, 山本一将 : 方策勾配法による局面評価関数とシミュレーション方策の学習, 情報処理学会研究報告, 2013-GI-30, No. 6, pp. 1 - 8 (2013).

「方策勾配を用いた教師有り学習による
コンピュータ大貧民の方策関数の学習と
モンテカルロシミュレーションへの利用」
正誤表

| 訂正箇所 | 誤 | 正 |
|---|---|---|
| 第 3 頁 3.2 節 6 行目 | 西野 [6] | 西野ら [6] |
| 第 5 頁 表 3 | $\ln(\frac{\text{着手後手札の平均階級}}{\text{着手前手札の平均階級}}) \times$ 着手後残り手札枚数 | $\ln(\frac{\text{着手後手札の平均強さ}}{\text{着手前手札の平均強さ}}) \times$ 着手前残り手札枚数 |
| 第 5 頁 表 3 | 8 切りを出すとき，元々の自分の手札枚数とその 2 乗 | 8 切りを出すとき，着手後の自分の手札枚数とその 2 乗 |
| 第 5 頁 式 (3) | $L(\pi_*, \pi_\theta) = -\pi_\theta(s, x)$ | $L(\pi_*, \pi_\theta) = -\ln \pi_\theta(s, x)$ |
| 第 5 頁 式 (4) | $\theta \leftarrow \theta + \frac{\alpha}{T}[\phi(s, x) - \sum_{b \in A} e^{\phi(s,b)\cdot\theta}]$ | $\theta \leftarrow \theta + \frac{\alpha}{T}[\phi(s, x) - \sum_{b \in A} \pi_\theta(s, b)\phi(s, b)]$ |
| 第 6 頁 6 節<br>13 〜 15 行目 | 機械学習を用いているため<br>ライト級の出場要件を満たさないが，<br>計算量はライト級の出場プログラムと同程度である | 残り 2 人時の全探索などを行うが、<br>それでも計算量は対戦相手プログラムと同程度であり，<br>2015 年大会ライト級の出場要件を満たしている |