The Japan Society of Mechanical Engineers

2B1-4

強化学習を用いたコンピュータ将棋における状態表現に関する考察

State Representation of Reinforcement Learning for Shogi

○今津拓哉(岡山大)　半田久志(岡山大)
阿部匡伸(岡山大)

Takuya IMADU, Okayama University
Hisashi HANDA, Okayama University
Masanobu Abe, Okayama University

Recently, evaluation functions for Shogi by using computer has attracted much attention due to Bonanza based on machine learning. The Bonanza has achieved one of the strongest computer players for Shogi, which often defeat human players. In order to learn the evaluation functions, Bonanza utilizes a considerable number of game records. Meanwhile, reinforcement learning can learn evaluation values based on experiences. The reinforcement learning, however, has not succeeded in learning with a large number of fine-grained feature values. In this paper, we investigate the effects of the state representations in the evaluation functions for learning results, where the state representations are derived from the ones of 'Bonanza'.

Key word: Shogi, Reinforcement Learning, Bonanza

1　はじめに

将棋は二人零和有限確定完全情報ゲームに分類されるため問題の定式化が容易である．このため，将棋を指すプログラム，いわゆるコンピュータ将棋は，人工知能の一分野として研究が進められてきた．コンピュータ将棋界ではすでに様々な将棋ソフトが開発されており，年々その強さは増している．

コンピュータ将棋のプログラムは，コンピュータチェスの手法を真似て盤面評価探索による方法が採用されている．この方法では，数手先の手まで読み，最も評価の高い盤面へ導く手を打つ．この評価の算出法を評価関数と呼び，評価関数の作り方と何手先まで評価の対象とするかでコンピュータ将棋の強さが決まる．

特に近年では，計算資源の増大に伴い，評価関数の重みを計算機により自動的に獲得する方法が注目されている．その代表的な手法として，機械学習を用いたBonanzaがある[1]．Bonanzaは，多量の棋譜情報から数万もの膨大な数の特徴ベクトルを生成して，評価関数を構成する仕組みになっており，プロ棋士に匹敵するほどの将棋ソフトとなっている．しかし，Bonanzaでは学習事例として使う棋譜情報を正事例として学習に用いるため，評価関数の精度は棋譜情報の品質に依存し，また，多量の棋譜情報の収集には労力がかかる．

その一方，経験に基づいた学習方法として，強化学習がある．この手法では，学習するために棋譜情報などの教師データを必要とせず，何度も対戦を繰り返した経験から評価関数を構築することができる．実際に，強化学習により評価関数の重みを自動的に最適化する研究は古くから行われており，様々な方法でその重みは表現されてきた．

本発表では，この強化学習の枠組みにおいて，評価関数の状態表現の違いによる学習結果の差について考察する．

2　強化学習

2.1　TD学習について

強化学習の代表的な手法として，TD学習がある．TD学習は環境のダイナミクスのモデルを用いずに，経験から直接学習することができる手法である．また，近い将来の評価値の変化を学習の基準とするため，最終結果を待たずに，パラメータを更新することができる[2]．

本研究では，このTD学習を用いて学習実験を行う．

2.2　TD学習の将棋への適用

今回の学習では，以下の式を用いて，1手打つごとに評価関数の重みを更新していく[3]．

$$W_{t+1} = W_t + \alpha(P_{t+1} - P_t)\sum_{i=1}^{t}\lambda^{t-i}\nabla_W P_i \qquad (1)$$

ここで，W_tはt手目の局面における評価関数の重みベクトル$W=(w_1,w_2,\ldots,w_i,\ldots,w_n)$で，ベクトルの要素$w_i$は$i$番目の特徴の価値を表している．本発表では学習により重みベクトルを定める：αは学習レートを制御するパラメータ(学習率)，λは$0\leq\lambda\leq1$の範囲を取り，過去の状態を学習に反映させる程度を制御するパラメータ(割引率)である．そして，$\nabla_W P_i$は各パラメータw_iによる勾配ベクトルを表し，

$$\nabla_W P_i = (\partial P_i/\partial w_1, \partial P_i/\partial w_2, ..., \partial P_i/\partial w_n) \qquad (2)$$

と定義する．さらに，P_iはその局面からそのゲームが勝利に終わる予測確率で，以下のシグモイド関数で表す．

$$P(E(K)) = 1/(1+e^{-E(K)/1000}) \qquad (3)$$

ここでの$E(K)$は局面Kでの評価関数を用いて算出した評価値のことで，以下のように表す．

$$E(K) = \sum_{j=1}^{N} w_j x_j(K) \qquad (4)$$

$x_j(K)$は局面Kにおける特徴jの要素の特徴量である：駒価値の場合は，(先手側の駒jの枚数) − (後手側の駒jの枚数)とする．本発表で用いた王将との相対位置で定義される状態では，特徴jに関連付けられた王将と他の駒の位置関係が観測された場合は1，観測されない場合は0とする．

3　評価関数の状態表現

ボードゲームの学習において，より強力な思考プログラムを作るためには，その学習手法と共に評価関数を構成する重みの表現方法も重要な要素である．これまで，評価関数を表すために，駒価値や王と駒との距離，王の安全度など，様々な種類の重みが用いられてきた[3]．

NII-Electronic Library Service

The Japan Society of Mechanical Engineers

そこで，今回は次の3つの要素に着目して評価関数を構成することにした．

・駒価値

・2つの王と他1の駒との位置関係

・王と他の2駒との位置関係

ここで，駒価値は，王将から歩，そして，各成駒について，別々に定義されている（計14個の重み要素となる）．つまり，前節の価値関数を構成する重みベクトルで，別々の要素w_jとなる．また，王将との相対位置として，「2つの王と他の1駒との位置関係」と「王と他の2駒との位置関係」の二つのを用いた．

「2つの王と他の1駒との位置関係」では，王の位置の組合せが$81 \times 80 = 6480$通りである．他の1駒は先手側の駒のみ考え（後手側は反転して考える），その駒の種類が9種類（金と成り金の区別はしない）あり，盤面上の位置を考慮すると，$9 \times 81 - (4 \times 9) = 693$通りの組合せがある．（歩や香車や桂馬では存在できないマスがあるため，その分減らす．）さらに，各持ち駒の枚数(枚数0の場合も含む)を考慮に入れると，$18 + 1 + (4 + 1) \times 4 + (2 + 1) \times 2 = 45$通りとなり，これらを合計すると，他の1駒は$684 + 45 = 738$通りとなる．よって，2つの王と他の1駒との位置関係では，$6480 \times 738 =$約480万種類の特徴がある．

同様に，「王と他の2駒との位置関係」では，王の位置は先手側だけ考慮し，81種類．王以外の駒は先手側と後手側を区別して考えるため，$738 \times 2 = 1476$種類となる．また，他の2駒の位置は，重複する場合を除くと，その組み合わせは$1476 \times (1476 + 1)/2 =$約110万通りとなる．（その中には他2駒が同一の駒になる場合があり，これは王と他1駒の位置関係と表すこともできる．）よって，王と他の2駒との位置関係では，110万$\times 81 =$約8900万種類の特徴となる．

これらを全て合わせると，約1億個の特徴となり，それを式(4)で計算したものが今回の評価関数である．これは，Bonanza(version 4など)で用いられている評価関数表現に近いものとなっている．Bonanzaでは多量の棋譜を用いて学習させることにより，その評価関数を構成している[4]．

本実験では，強化学習によりこれらの要素を学習させ，更には，それらの比較をすることにより，強化学習における評価関数の状態表現の違いによる学習結果の差を調べる．各状態において，「2つの王と他の1駒との位置関係」では，盤面に存在している駒の個数とほぼ同数の特徴が重み更新の対象となる．「王と他の2駒との位置関係」では，盤面に存在している駒の個数の自乗とほぼ同数の特徴が重み更新の対象となる．

4 実験

4.1 実験手法

本実験で用いた将棋システムは以下の特徴を有する：

・$\alpha\ \beta$法を用いた全幅探索

・3手詰めの詰め将棋ルーチン

・前節で示した評価関数

・探索の深さは3

・Bonanzaに付属されていた定跡の使用

これは，将棋ソフトBonanzaのシステムに近く，本研究ではBonanzaのプログラムに手を加えた．具体的には，Bonanza(Feliz)に，2節で述べたようなTD法による学習システムを組み込み，実験を行った．

	相手1	相手2	相手3
実験1-1	39勝41敗20分		50勝35敗15分
実験1-2	46勝38敗16分	43勝39敗18分	
実験1-3	49勝25敗26分		

Table 1: Evaluation after learning: Experiment-1

また，その際$\lambda = 0.8$，$\alpha = 100 \sim 1$へと徐々に下がっていくものとし，重みの学習は学習者の手番が来るごとに行う．実験は評価値を学習するプログラムを先手，対戦相手を後手とし，学習開始前の各学習パラメータは全て0とし，評価値に乱数を加えて，同じ手順の対局を繰り返さないようにした．

以上の設定で，本発表では次の2つの実験を行う：

実験1 対戦相手をBonanzaとして10000局の学習を3回行う．その際，学習プログラムにおける評価関数の状態表現を変えて，それぞれ異なるパラメータを学習させる．

実験2 学習プログラムの評価関数の状態表現は全て同じで，10000局の学習を3回行う．その際，対戦相手をそれぞれ変更して学習させる．

4.2 実験1

まず，対戦相手をもともとのBonanza(王と他の2駒との位置関係，2王と他1駒の位置関係，駒価値で判断するプログラム)として10000局学習行う．

その際，学習プログラムの評価関数の状態表現は以下の通りである．状態数の変化に伴い，学習させるパラメータの個数も増減する：

実験1-1 駒価値のみ

実験1-2 2王と他1駒の位置関係，駒価値

実験1-3 王と他の2駒との位置関係，2王と他1駒の位置関係，駒価値

この3種類の学習実験を行った後，学習後のプログラムとBonanzaなどを100局対戦させて評価する．この評価時の対戦相手として，Bonanzaに基づいた，次の3つの相手を用意した：

相手1 もともとのBonanza(王と他の2駒との位置関係，2王と他1駒の位置関係，駒価値で判断するプログラム)

相手2 Bonanzaにおいて2王と他1駒の位置関係，駒価値で判断するプログラム

相手3 Bonanzaにおいて駒価値のみで判断するプログラム

実際に実験を行い，評価すると，表1のような結果になった．学習前の価値関数では，一度も勝てなかったことから，この結果より，本実験の枠組みで，将棋の政策を学習できたことが判る．

それぞれの実験において学習後のものは，同じ評価関数の状態表現を使用したものと対戦させた場合，5割近く勝つことができた．また，それぞれをBonanzaと対戦させた場合でも，ある程度勝ち越すことができた．

今回の実験では，評価関数のパラメータ数が多い実験1-2，実験1-3でも上手く学習できており，Bonanzaと対戦させた時も，評価関数のパラメータ数が多いほど勝利数が多くなっている．

NII-Electronic Library Service

The Japan Society of Mechanical Engineers

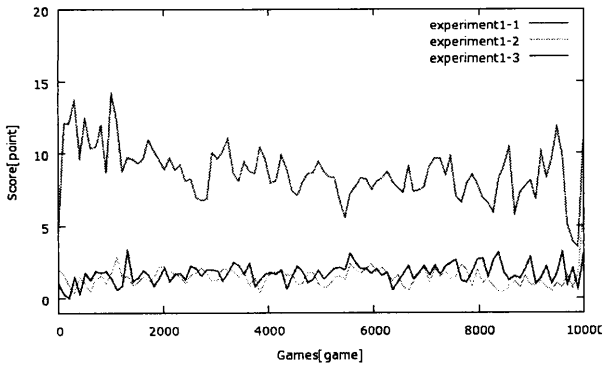

Fig. 1: Learning progress of experiment-1

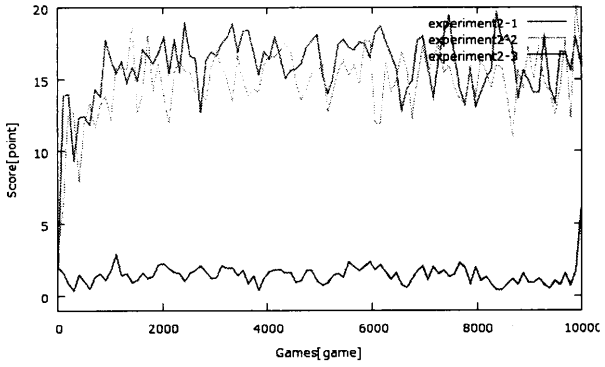

Fig. 2: Learning progress of experiment-2

このことから，今回のようにBonanzaを利用して強化学習をした場合，パラメータ数が多くても上手く学習できることが分かった．

また，図1は学習の様子である．x局目において，$x-10 \leq x < x+10$局目における学習時の勝敗について，$score$ = 勝利数 − 敗北数 + 20を算出し，y軸に用いている．したがって，勝率が5分の時，$score$は20付近の値を示し，全敗の時は0となる．学習時には評価値に乱数を加えているため，学習時の勝敗が直接強さに結びつくことはないが，ある程度の学習の経過は分かると考えている．

4.3 実験2

実験2として，学習プログラムの状態表現を固定し，学習時の対戦相手の評価関数の状態表現を変更して学習を行った．具体的には，学習プログラムの状態表現を2王と他1駒の位置関係，駒価値とした．そして，対戦相手の評価関数の状態表現は次の通りにした．

実験2-1 駒価値のみ

実験2-2 2王と他1駒の位置関係，駒価値

実験2-3 王と他の2駒との位置関係，2王と他1駒の位置関係，駒価値

実験2-1から2-3までの対戦相手のそれぞれに対して，10000局対局させて学習した．

学習後のプログラムを実験1の相手2と100局対戦させて評価する．結果を表2にまとめる：この結果を見ても分かるように，学習する際の対戦相手を変化させた場合も学習結果に違いが出た．このことから，学習時の対戦相手によって，学習したものも左右されるということが分かった．

今回の対戦相手である評価関数の表現が駒価値だけのもの(実験2-1)，2王と他1駒の位置関係，駒価値のもの(実験2-2)，2王と他1駒との位置関係，王と他の2駒との位置関係，駒価値のもの(実験2-3)はそれぞれ対戦させると，評価関数が複雑な実験2-3のものが僅差で最も強くなる．

	相手2
実験2-1	53勝29敗18分
実験2-2	42勝32敗24分
実験2-3	48勝32敗20分

Table 2: Evaluation after learning: Experiment-2

しかし，それらを対戦相手として学習させたものは，実験2-1のものが最も強くなり，対戦相手が強いほど，また，評価関数が複雑なほど，強く学習できるとは必ずしも限らない．

なお，図2は実験2での学習の様子である．グラフのx軸，y軸は実験1と同じものを用いた．

5 おわりに

今回，コンピュータ将棋における評価関数の強化学習において，評価関数の状態表現を様々な形に変化させて学習する実験について述べた．

実験1の結果，学習させる評価関数の状態表現として，駒価値だけ用いたもの，また，2王と他1駒との位置関係や王と他の2駒との位置関係を加えたもののいずれも上手く学習することができた．

更に，実験2の結果，学習時の対戦相手によって，学習したものも左右されるということが分かった．加えて，今回の実験結果から，対戦相手が強いほど，また，評価関数が複雑なほど，強く学習できるとは必ずしも限らないことが推測される．

今回の学習実験では，評価関数のパラメータ数が非常に多かったため，上手く学習させる方法が問題となったが，(定跡を用いた)Bonanza(Feliz)に強化学習を組み込んで実験することで，上手く学習することができた．しかし，今回の結果はBonanzaのプログラムや定跡に依存している可能性があり，今後の再考が必要である．また，実験の結果は偶然性を持ち，今回の実験が完全に正しいと断言できないため，今後の更なる実験が必要である．

今後の課題としては，別の将棋プログラムでの実験による普遍性の向上，更なる実験による確実性の向上，別の評価関数の状態表現による学習，より効率的な強化学習手法の研究，対戦相手に依存するこのない学習方法の研究などが挙げられる．

参考文献

[1] 保木邦仁．将棋プログラム Bonanza． web, 2005. (http://www.geocities.jp/bonanza_shogi/).

[2] Richard S.Sutton, Andrew G.Barto(著)，三上貞芳，皆川雅章(訳)，「強化学習」，森北出版，第6章，2000.

[3] 薄井克俊，鈴木豪，小谷善行，"TD法を用いた将棋の評価関数の学習"，ゲームプログラミングワークショップ'99，pp.31-38，1999.

NII-Electronic Library Service

The Japan Society of Mechanical Engineers

[4] 保木邦仁. "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.

NII-Electronic Library Service