

コンピュータ将棋における方策勾配を用いた局面評価関数の教師付学習

大串明^{†1} 山本一将^{†2} 森岡祐一 五十嵐治一^{†1}

概要: 方策勾配を用いた教師付学習をコンピュータ将棋の局面評価関数の学習に適用する方法が考案されている。本論文ではこの学習法の性能評価実験を行ったので結果を報告する。

キーワード: コンピュータ将棋, 方策勾配, 教師付学習, 局面評価関数

Supervised Learning of Positional Evaluation Functions in Computer Shogi Based on Policy Gradients

AKIRA OHGUSHI^{†1} KAZUMASA YAMAMOTO^{†2}
YUICHI MORIOKA HARUKAZU IGARASHI^{†1}

Abstract: A supervised learning using policy gradients was proposed for learning positional evaluation functions in computer shogi. We conducted a preliminary experiment to verify the ability of the learning method and reported on the results in this paper.

Keywords: Computer Shogi, Policy gradient, Supervised learning, Positional evaluation function

1. はじめに

近年, コンピュータ将棋の棋力向上がめざましい。プロ棋士との対戦でも大きく勝ち越すようになってきた。この大きな理由の一つとして, 機械学習による局面評価関数の構築が成功したことが挙げられる。この代表例が, 「Bonanza メソッド」と称される教師付学習法である。

一方, 強化学習を用いた局面評価関数の学習を試みようとする研究の流れがある。すでに, チェスでは価値ベースの強化学習である, TD 法や TDLeaf(λ)法を適用した例[1]が報告されており, 将棋へも適用が試みられてきた[2][3]。しかし, 各局面の状態価値関数が定義できるのは報酬にマルコフ性がある場合に限られており, 一連の局面の内容や一局全体の内容の評価して報酬を与えるような場合には適用できない。そこで, 我々はこれまでに報酬のマルコフ性を必ずしも必要としない方策ベースの強化学習である方策勾配法を将棋へ適用することを提案してきた[4][5]。さらに, 価値ベースの強化学習と方策ベースの強化学習を VAPS アルゴリズムにより統一的な枠組みで取り扱おうとする融合方式を最近提案した[6]。

この融合方式では, 強化学習のみならず, 教師付学習をも同一の枠組みで取り扱うことを意図して, 「方策勾配を用いた教師付学習法」を提案している。本報告では, この学習法を実際の将棋対局譜を用いて簡単な学習評価実験を行ったので, その実験結果を報告する。これにより, この学習方法の有効性を検証するとともに今後の可能性を検討し

たい。

2. 学習方式

文献[6]にしたがって, 本研究で用いる教師付学習の方式を簡単に述べる。まず, 最小化すべき学習の誤差関数 $\delta_{KLD}(\sigma; \pi^*, \pi)$ を次のように定義する。

$$\delta_{KLD}(\sigma; \pi^*, \pi) \equiv \sum_{s \in \sigma} \sum_{a \in A(s)} \pi^*(a|s) \ln[\pi^*(a|s)/\pi(a|s; \omega)] \quad (1)$$

ただし, 教師となる着手決定方策を π^* , 学習システムの着手決定方策を π とする。両者ともに確率分布関数として与えられているとする。 $A(s)$ は局面 s における合法手の集合, ω は局面評価関数中のパラメータである。

(1)の $\delta_{KLD}(\sigma; \pi^*, \pi) (\geq 0)$ は, 正解の方策 π^* と学習システムの方策 π との距離を表すカルバック・ライブラー情報量 (Kullback-Leibler divergence) である。ここで, (1)の $\delta_{KLD}(\sigma; \pi^*, \pi)$ の勾配,

$$\nabla_{\omega} \delta_{KLD}(\sigma; \pi^*, \pi) = - \sum_{s \in \sigma} \sum_{a \in A(s)} \pi^*(a|s) \nabla_{\omega} \ln \pi(a|s; \omega) \quad (2)$$

には方策 π の勾配が含まれている。(2)から勾配法により, 学習則は

$$\Delta \omega = -\epsilon \nabla_{\omega} \delta_{KLD}(\sigma; \pi^*, \pi) \quad (3)$$

と表される。さらに, PGLeaf 法[4]と同じように, 方策

^{†1} 芝浦工業大学工学部情報工学科
Shibaura Institute of Technology

^{†2} 株式会社コスモ・ウェブ
Cosmoweb Co., Ltd.

$\pi(a|s; \omega)$ を、局面 s で手 a を指した後の最善応手手順の leaf 局面 s^* の評価値 $E_s(s^*|a, s; \omega)$ を用いた Boltzmann 分布で表す。

$$\pi(a|s; \omega) = \exp(E_s(s^*|a, s; \omega)/T)/Z \quad (4)$$

$$Z \equiv \sum_{x \in A(s)} \exp(E_s(s^*|x, s; \omega)/T) \quad (5)$$

T は温度と呼ばれるパラメータである。(4),(5)を(3)へ代入すると、

$$\Delta\omega = \varepsilon \sum_{s \in \sigma} \sum_{a \in A(s)} \pi^*(a|s) \nabla_{\omega} \ln \pi(a|s; \omega) \quad (6)$$

$$= \frac{\varepsilon}{T} \sum_{s \in \sigma} \sum_{a \in A(s)} \pi^*(a|s) \cdot [\nabla_{\omega} E_s(s^*|a, s; \omega) - \sum_{x \in A(s)} \pi(x|s; \omega) \nabla_{\omega} E_s(s^*|x, s; \omega)] \quad (7)$$

となる。これは次のように表される[6]。

$$\Delta\omega = \frac{\varepsilon}{T} \sum_{s \in \sigma} \sum_{a \in A(s)} \pi^*(a|s) \cdot [(1 - \pi(a|s; \omega)) \nabla_{\omega} E_s(s^*|a, s; \omega) - \sum_{x \neq a} \pi(x|s; \omega) \nabla_{\omega} E_s(s^*|x, s; \omega)] \quad (8)$$

ここで、[]内の2つの項の符号を考える。第1項では、 $1 - \pi(a|s; \omega) > 0$ なのでエピソード中に学習エージェントが実際に指した手 a の価値 $E_s(s^*|a, s; \omega)$ を高める方向に ω は更新される。第2項では、 $-\pi(x|s; \omega) < 0 (x \neq a)$ なので a の兄弟手 x の価値を低下させる方向に ω は更新されることが分かる。特に、棋譜からの学習のように正解手が1つに限定される場合には、正解手の価値を高め、兄弟手の価値を低下させる方向に ω が更新されることを表している。

また、正解手が1つでなく、確率分布で与えられる場合には、(7)を次のように変形すると意味がわかりやすい。

$$\Delta\omega = \frac{\varepsilon}{T} \sum_{s \in \sigma} \sum_{a \in A(s)} [\pi^*(a|s) - \pi(a|s; \omega)] \nabla_{\omega} E_s(s^*|a, s; \omega) \quad (9)$$

(9)の右辺の[]内の符号に注意すると、局面 s において正解分布の選択確率値よりも学習中の方策の選択確率値が小さい場合にはその指し手 a の価値を高め、逆の場合は過大評価なので価値を低下させようとしているのが分かる[6]。

現在のコンピュータ将棋において主流になっている「Bonanza メソッド」も教師付学習であるが、(9)の学習法は、①確率分布同士の比較なので、正解手の頻度分布が既知であればそれを利用することが可能である点と、② ω の絶対値が増大することを防止する正則化項が必ずしも必要でない点が相違点としてあげられる。

3. 学習実験

3.1 実験概要

プロ棋士の対局を記録した棋譜データベースから教師用データ（訓練用データ）を作成する。その対局面を学習

システムへ順に見せ、(9)による局面評価関数中のパラメータ ω を更新する。今回は、一試合ごとではなく、一局面ごとにパラメータを更新するオンライン的な学習を行う。この学習過程におけるプロ棋士の着手とシステムの着手との一致率の推移を調べる。

3.2 教師データの作成

学習実験に用いる教師データとして、インターネット上のサイトである「2chkifu」[7]上に掲載されている 55,800 局の対局譜に含まれている局面 s とその局面での着手 a の組 (s, a) を用いた。この棋譜データベースには、プロ棋士同士の公式戦の他、朝日アマ将棋名人戦などのアマチュア棋士の棋譜も含まれている。

3.3 評価関数と学習パラメータ

局面評価関数として、Bonanza 6.0 の評価関数を用いた[8]。すなわち、駒割り（各駒に固有の点数を与えたもの）と 3 駒関係（2 玉と他の 1 駒の位置関係「KKP」と、玉と他の 2 駒の位置関係「KPP」）を特徴量とする線形関数である[9]。

学習実験では、駒割の値は Bonanza の値に固定し、3 駒関係の重み係数を学習対象とした。なお、学習パラメータの初期値はすべて 0 とした。

3.4 探索プログラムと深さ

学習時の探索プログラムとして、Bonanza 6.0 を用いた。ただし、今回は探索の深さは 1 とした。すなわち、対象局面から 1 手指した局面の局面評価関数の値をその手の評価値とした。この際、静止探索なども全く行っていない。

3.5 実験結果

(7)に示した学習則を用いて、学習率 ε と温度 T を次の 5 つの場合に設定し、それぞれ学習実験を行った： $(\varepsilon, T) = (100, 1), (100, 10), (10, 1), (10, 10), (1, 1)$ 。結果を図 1 に示す。ただし、 $\pi^*(a|s)$ は、局面 s についての着手 a の分布ではなく、訓練用データ (s, a) ごとに教師の指し手ならば 1、それ以外の指し手ならば 0 の 2 値で与えた。また、55800 局の学習を 1 回の学習と定義する。図 1 には学習 1 回分の一致率の推移が記されている。なお、学習実験には Intel Core i7-3770 3.40GHz、メモリ 4GB の PC を使用したが、約 98 時間を行った。

図 1 の結果から、学習を 100 回行った場合、 $\varepsilon = 100, T = 1$ の時の学習率が最も高かった。この場合、未学習状態の一致率は約 15.7%程度であったが、100 回の学習終了後には約 44.2%まで向上している。したがって、(7)の学習則の学習能力を検証することは一応できたと言える。

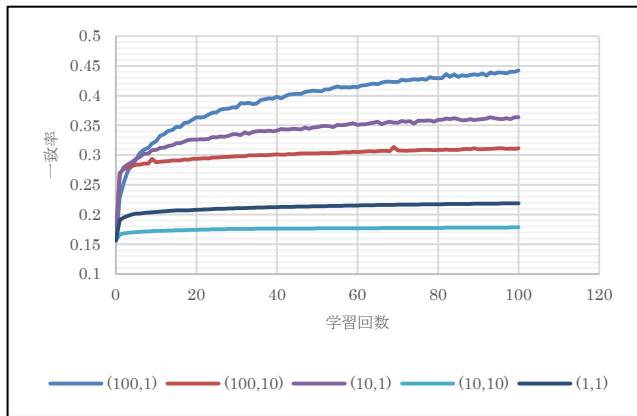
図1 各 (ϵ, T) における一致率の推移.

Figure 1 Change of the accuracy rates.

3.6 評価実験

学習後の学習パラメータを評価するために Bonanza6.0 同士の対局を行った. 2つの Bonanza6.0 のプログラムを用意し, 一方には学習済み ($\epsilon=100$, $T=1$, 学習回数 100) のパラメータを持つ評価関数を使用させ, もう一方には未学習のパラメータ (駒割は学習済みと同一で, それ以外は 0) で対局させた. この際, 定跡は不使用とし, それ以外の探索深さや時間制御, 枝刈り方法など探索部は公開されているプログラムをそのまま使用した.

未学習パラメータを使用したプログラムを先手とし, 学習済みパラメータを使用したプログラムを後手とした対局を行った. 持ち時間は 10 分 + 秒読み 10 秒である. 途中図を図 2 (13 手目 ▲ 2 二歩成まで) と図 3 (52 手目 △ 2 三銀まで) に示す. これから分かるように, 先手は駒割しか局面評価関数に用いていないので, 駒得を最優先として飛車先を伸ばし, 角頭を攻め, 序盤で角得して優勢に立ち (図 2), そのまま押し切っている. それに対し, 後手は王の囲いを優先し, 美濃囲いを完成させることを優先している (図 3). これは先手には見られない現象であり, 3 駒関係のパラメータを学習することにより, このような美濃囲いなどの将棋の「型」を習得することができたのではないかと考えられる.

なお, 棋譜は以下の通りである: ▲ 1 六歩 △ 3 二飛 ▲ 2 六歩 △ 6 二玉 ▲ 2 五歩 △ 7 二玉 ▲ 2 四歩 △ 同歩 ▲ 同飛 △ 8 二玉 ▲ 2 三歩 △ 1 四歩 ▲ 2 二歩成 (図 3) △ 同銀 ▲ 2 八飛 △ 7 二銀 (図 4) ▲ 2 四角 △ 2 三歩 ▲ 3 五角 △ 1 五歩 ▲ 5 三角成 △ 1 六歩 ▲ 4 三馬 △ 1 七歩成 ▲ 同香 △ 1 六歩 ▲ 同香 △ 同香 ▲ 同馬 △ 1 三香 ▲ 1 四歩 △ 同香 ▲ 1 五歩 △ 同香 ▲ 同馬 △ 5 二飛 ▲ 4 八銀 △ 5 五飛 ▲ 1 四馬 △ 5 四飛 ▲ 2 四歩 △ 8 四飛 ▲ 2 六歩 △ 3 二金 ▲ 1 五歩 △ 1 三歩 ▲ 2 三歩成 △ 1 四歩 ▲ 3 二と △ 2 七歩 ▲ 1 八飛 △ 2 三銀 (図 2) ▲ 同香成 △ 8 七飛成 ▲ 2 一と △ 6 四歩 ▲ 7 六銀 △ 8 四龍 ▲ 1 四歩 △ 6 五歩 ▲ 1 三歩成 △ 1 七歩 ▲ 3 八飛 △ 9 四歩 ▲ 7 五金 △ 8 六龍 ▲ 8 五金 △ 1 八歩成 ▲ 同飛 △ 7 六龍 ▲ 同歩 △ 1 七歩 ▲

同飛 △ 3 四歩 ▲ 6 四香 △ 6 六歩 ▲ 6 一香成 △ 6 七歩成 ▲ 6 二成香 △ 6 一銀打 ▲ 同成香 △ 同銀 ▲ 3 二飛 △ 6 二歩 ▲ 8 四歩 △ 6 五角 ▲ 8 三歩成まで 87 手で先手の勝ち.



図2 未学習プログラムと学習済みプログラムの対局 (13 手目 ▲ 2 二歩成まで).

Figure 2 Game between the programs with and without learning parameters (up to the 13th move).



図3 未学習プログラムと学習済みプログラムの局面図 (52 手目 △ 2 三銀まで).

Figure 3 Game between the programs with and without learning parameters (up to the 52th move).

4. まとめ

本報告では, 文献[6]で提案された「方策勾配を用いた教師付学習法」を Bonanza6.0 に実装し, 深さ 1 の探索レベルで学習実験を行った. 訓練用データとしてプロ棋士の棋譜 55800 局を用い, 駒の重みは固定した結果, 教師データとの一致率は約 15.7%程度から 100 回の学習終了後には約

44.2%まで向上した。また、学習後の対局内容から、美濃囲いなどの型を学習できたことがわかった。

今後は、学習時の探索の深くすることや、静止探索なども考慮した学習実験を行う予定である。この際、学習時間の増大が予想されるので、局面評価関数の対称性を利用するなどの計算の高速化を検討していきたい。

謝辞 本研究はJSPS 科研費 26330419 の助成を受けました。ここに感謝の意を表します。

参考文献

- 1) Baxter, J., Tridgell, A., and Weaver, L., : KnightCap: A chess program that learns by combining TD(λ) with game-tree search, Proceedings of the Fifteenth International Conference (ICML '98), pp.28-36 (1998).
- 2) Beal, D. F. and Smith, M. C.: Temporal difference learning applied to game playing and the results of application to shogi, Theoretical Computer Science, Vol. 252, pp.105-119 (2001).
- 3) 薄井克俊, 鈴木豪, 小谷善行: TD 法を用いた評価関数の学習, 第 4 回ゲーム・プログラミングワークショップ, pp.31-38 (1999).
- 4) 森岡祐一, 五十嵐治一: 方策勾配法と $\alpha \beta$ 探索を組み合わせた強化学習アルゴリズムの提案, 第 17 回ゲーム・プログラミングワークショップ, pp.122-125 (2012).
- 5) 五十嵐治一, 森岡祐一, 山本一将: 方策勾配法による静的局面評価関数の強化学習についての一考察, 第 17 回ゲーム・プログラミングワークショップ, pp.118-121(2012).
- 6) 五十嵐治一, 森岡祐一, 山本一将: プロ棋士の棋譜データベースを用いない局面評価関数の学習法についての考察, 第 34 回ゲーム情報学研究発表会(2015.7.4, 福岡市), 情報処理学会研究報告, Vol. 2015-GI-34, No. 4, pp.1-8 (2015).
- 7) 棋譜データベース作成プログラムの使い方, http://www.geocities.jp/shogi_depot/2chkifu.htm
- 8) Bonanza - The Computer Shogi Program, http://www.geocities.jp/bonanza_shogi/
- 9) 保木邦仁: 「Bonanza 4.1.3」のソースコード, コンピュータ将棋の進歩⑥ (松原仁 編著), 共立出版, pp.1-23, 2012 年.