Minimax 探索最適化の関数形

保木邦仁[*]　金子知適[**]

将棋において局面評価関数の機械学習が成功を収めている。この学習法では、局面評価関数の特徴ベクトル $\boldsymbol{v}$ を調整して minimax 探索を棋譜データに沿って最適化する。本研究では、この最適化に用いられる関数 $J(\boldsymbol{v})$の形状を、連続性、偏微分不可能性、極小点をどの程度持つかという点に焦点を絞り解析し、結果を報告する。

Landscape for optimization problem of minimax search

Kunihito Hoki* and Tomoyuki Kaneko**

In Shogi, machine learning of the positional evaluation function has proved successful. The learning method adjusts the feature vector $\boldsymbol{v}$ of the positional evaluation function by solving an optimization problem of minimax search in accordance with professional game scores. In this paper, we analyze properties of the function $J(\boldsymbol{v})$ to be optimized in terms of continuity, partial differentiability, and number of local minima.

1. はじめに

2005 年以降、コンピュータ将棋において局面評価関数の機械学習が注目されている。近年、この技術は強いコンピュータ将棋を作成するための必須技術となっており、主要な商用将棋プログラムはこの関数の大規模な自動学習を行なっている。また、このゲームプログラミングワークショップにおいて多数の関連研究が報告されている [1-6]。

*電気通信大学先端領域教育研究センター
Center for Frontier Science and Engineering
University of Electro-Communications
hoki@cs.uec.ac.jp

**東京大学大学院総合文化研究科
Graduate School of Arts and Science
University of Tokyo
kaneko@graco.c.u-tokyo.ac.jp

この機械学習法では、ゲーム局面 p の評価関数 $f(p,\boldsymbol{v})$を特徴づけるベクトル $\boldsymbol{v}$ を調整するために、minimax 探索を棋譜データに沿って最適化する。この最適化に用いられる関数（以下、目的関数と呼ぶ）は以下のように書かれる。

$$J(\boldsymbol{P},\boldsymbol{v})=\sum_{p\in\boldsymbol{P}}\sum_{m=2}^{M_p}T_p\left[\xi(p_m,\boldsymbol{v})-\xi(p_1,\boldsymbol{v})\right]+\lambda g(\boldsymbol{v})+L(\boldsymbol{v}) \tag{1}$$

ここで、$\boldsymbol{P}$ は棋譜データに現れる局面集合、M_p は局面 p の合法手数、p_1 は棋譜で選択された子局面、p_m は m 番目の合法手による子局面、$\xi(p_m,\boldsymbol{v})$ はゲーム木探索の値である。式(1)右辺第 2 項目はラグランジュ未定乗数λにより導入された拘束条

件 $g(\boldsymbol{v}) = 0$ であり、これによって駒割り等の主要な特徴要素の大きさを拘束した最適化が行なわれる。第 3 項目は正則化項であり、L_2 正則化の場合には $L(\boldsymbol{v}) = \|\boldsymbol{v}\|$となる。

損失関数 $T_p(x)$ は、局面 p の手番が Max プレイヤーの場合 $T_p(x) = T(+x)$、Min プレイヤーの場合 $T_p(x) = T(-x)$ であり、$T(x)$ は一価の単調増加関数とする。$T(x)$ を階段型関数にとると、$J(\boldsymbol{P},\boldsymbol{v})$ はサンプルされた全局面中、将棋プログラムが誤判断をしてしまう合法手の総数となる。従って、「プログラムの探索結果を棋譜と同じにする」という目標は「目的関数 $J(\boldsymbol{P},\boldsymbol{v})$ を最小にする特徴ベクトル $\boldsymbol{v}$ の発見」という最小化問題に置き換えられる。2006 年に報告された手法では、損失関数 $T(x)$ にシグモイド関数 $(1 + e^{-x})^{-1}$ が用いられている [1]。2008 年に報告された手法では、ロジスティック回帰に対応する関数 $T(x) = \log(1 + e^{-x})$ が用いられている [5]。将棋以外のゲームでは、チェスにおいてこの手法と同じ枠組みの機械学習法が 2001 年に報告されているが [7]、実用的な大規模自動学習には至っていない。

本論文では、コンピュータ将棋の学習において重要な役割を果たす目的関数 $J(\boldsymbol{P},\boldsymbol{v})$の関数形について考察する。この最適化に用いられる関数は、1 万以上の棋譜から派生する多数局面の minimax 値に依存することから、複雑な関数形を持つことが予想される。そこで、将棋プログラムにおいて $J(\boldsymbol{P},\boldsymbol{v})$を最小化する特徴ベクトル $\boldsymbol{v}$ を数値的に計算するために必要な知識となる以下のこと、

- $J(\boldsymbol{P},\boldsymbol{v})$は連続か
- $J(\boldsymbol{P},\boldsymbol{v})$の偏微分不可能点の性質
- $J(\boldsymbol{P},\boldsymbol{v})$は極小点をどの程度持つか

に注目する。

2. ゲーム木探索が正確な minimax 値を返す等の理想的な場合

探索深さを一定基準で制限し、末端の値に局面評価関数を用いるゲーム木探索を考える。ここで、プログラムの探索結果は正確な minimax 値を与えるとする。また、簡単のために式(1)中の束縛条件の関数 $g(\boldsymbol{v})$, 正則化関数 $L(\boldsymbol{v})$, 損失関数 $T(x)$、及び局面評価関数 $f(p,\boldsymbol{v})$ は連続で（偏）微分可能と仮定する。このような理想的な状況で、目的関数 $J(\boldsymbol{P},\boldsymbol{v})$の連続及び偏微分可能性は、minimax 探索の値$\xi(p,\boldsymbol{v})$の性質のみで決まる。

まず、$\xi(p,\boldsymbol{v})$が $\boldsymbol{v}$ に対して連続であることを示す。$\xi(p,\boldsymbol{v})$が minimax 探索の値であることから不等式$|\xi(p,\boldsymbol{v}+\boldsymbol{\Delta}) - \xi(p,\boldsymbol{v})| \le \mathop{\mathrm{Max}}_{p'\in\Im} |f(p',\boldsymbol{v}+\boldsymbol{\Delta}) - f(p',\boldsymbol{v})|$ が成り立つ。但し、$\Im$ は p をルートとする minimax 木の末端局面集合、$\boldsymbol{\Delta}$はベクトル $\boldsymbol{v}$ の変位ベクトルである。この不等式は、minimax 木の末端から数学的帰納法を適用することにより証明される。また、すべての $p' \in \Im$ において $f(p',\boldsymbol{v})$が $\boldsymbol{v}$ に対して連続という仮定から、任意の正数$\varepsilon > 0$ に対して$\delta > 0$ が存在し、$|\boldsymbol{\Delta}| < \delta$ ならば $\mathop{\mathrm{Max}}_{p'\in\Im} |f(p',\boldsymbol{v}+\boldsymbol{\Delta}) - f(p',\boldsymbol{v})| < \varepsilon$となる。従って、$|\xi(p,\boldsymbol{v}+\boldsymbol{\Delta}) - \xi(p,\boldsymbol{v})| < \varepsilon$ であり、$f(p,\boldsymbol{v})$が $\boldsymbol{v}$ に対して連続であるなら$\xi(p,\boldsymbol{v})$も連続であることが示された。

一方、$\xi(p,\boldsymbol{v})$はベクトル $\boldsymbol{v}$ の要素 v_i に関して偏微分可能とは限らない。偏微分不可能点では、その点で $v_i = c$ として、v_i が $c - \Delta$から $c + \Delta$に変化し最善応手系列（PV）が変更される。また、この偏微分不可能な点では、PV に対応する末端局面が 2 つ以上存在する。偏微分可能な場合には、導関数は以下のように書かれる。

$$\frac{\partial \xi(p,\boldsymbol{v})}{\partial v_i} = \frac{\partial f(p^{\mathrm{leaf}},\boldsymbol{v})}{\partial v_i} \tag{2}$$

ここで、p^{leaf} は PV に対応する単一の末端局面で

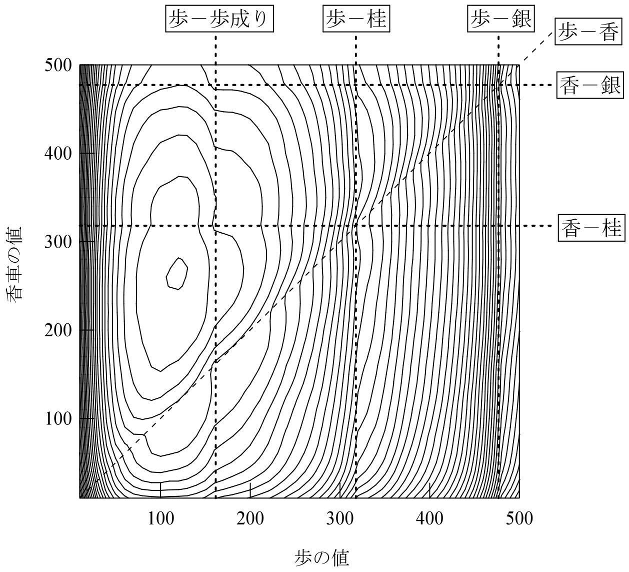

図 1　将棋の駒価値学習における目的関数 $J(\boldsymbol{P},\boldsymbol{v})$ の歩・香車面の等高線。他 11 種類の駒価値は最適解付近の値、駒価値以外の $\boldsymbol{v}$ の要素はすべて 0、拘束条件や正規化は用いていない。破線は図中ラベルに示される 2 種の価値が等しくなる位置を表す。

ある。

3. 将棋プログラムと棋譜データを用いた現実的な場合

本節では、実際の将棋プログラムを使って目的関数を計算し、その形を調べる。前節の理想的な状況とは異なり、現在の将棋プログラムのゲーム木探索は minimax 値を与えない。実現確率探索 [8]、Late Move Reduction (LMR) [9], Futility Pruning [10, 11], 遷移局面表などの手法を利用して探索の効率化を図っているからである。さらに、局面評価は整数型の変数により行われる。

図 1 に Bonanza を用いて求めた目的関数の歩・香車 2 次元面を示す [12]。他 11 種類の駒価値に関しては最適解付近の値、318（桂）、477（銀）、562（金）、620（角）、734（飛）、485（と）、387（成香）、445（成桂）、343（成銀）、781（馬）、957（竜）を用いた。図中の等高線は、歩・香価値それぞれ 10 点おきに目的関数値を求め計算した。

このプログラムは、損失関数 $T(x)$にシグモイド関数を採用している。また、探索の値$\xi(p,\boldsymbol{v})$は基準深さ 1 で求め、棋譜データはプロ棋士等の 1,000 局を用いた。計算により求められた目的関数面は、おおむね連続であった。Bonanza に用いられている LMR や futility pruning などの影響により不連続点が生じる可能性があるが、そのような現象は駒価値を数百程度の整数で表す計算精度では確認することはできなかった。また、前節で説明された偏微分不可能点は、実際の将棋プログラムによる計算によっても確認された。2 次元面上ではこの点は線状に存在する様子が図 1 破線により示されている。偏微分不可能点は、いずれか 2 種類の駒を取る価値や成る価値の大きさが等しくなる位置に存在していた。

図 1 には示されていないが、歩・香車の価値が 10 から 1,000 の範囲において極小点は 1 つしか存在しない。より複雑な局面評価関数を用いた場合での極小点数を確認するために、公開されている Bonanza の評価関数をそのまま用いた実験をおこなった。この計算では、オリジナルの特徴全要素に標準偏差 22 程度の正規乱数を加えて、6 通り(No. 1 ~ 6)の初期値を用意した。最適化の拘束条件は駒価値の総和に課し、L_1 正規化を用いた。偏微分値は、プログラムの返す 1 つの PV 末端局面を用いて近似計算した。2 式(1)右辺の $g(\boldsymbol{v})$や $L(\boldsymbol{v})$の具体的な関数形及び最適化手法の解説は紙面の都合上割愛するが、具体的な計算手続きや使用する定数は公開されているソースコードそのままに用いた。但し、計算時間短縮のため、探索の値$\xi(p,\boldsymbol{v})$を計算する基準深さを 3 から 2 に変更した。また、最適化計算の 1 回の反復あたりの基準ステップ数は 32 として、約 5 万局を棋譜データとして使用した。

図 2 に、この 6 通りの初期値を用いた目的関数 $J(\boldsymbol{P},\boldsymbol{v})$の最適化結果を示す。この図から、反復回

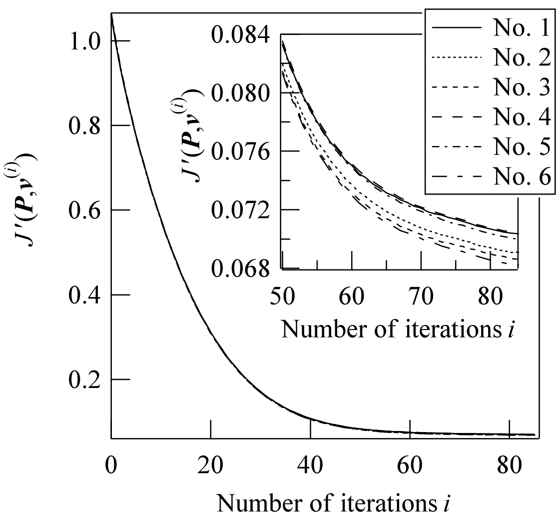

図2　6 つの異なる特徴ベクトル $\boldsymbol{v}$ を初期値とした最適化の結果。

数 i が増えるほど、それぞれの関数値のばらつきは大きくなる様子がわかる。しかし、このばらつきの幅は $i = 80$ 付近でも 0.002 程度であり、これは初期値からの関数値の減少分と比較すると無視できるほど小さい。この結果は、極小点が複数存在するが、各局所解に対応する局面評価関数の性能は大体どれも同じであるということを示唆している。確認のため、対局を計 3,000 回以上行い 6 つの最適化された特徴ベクトルのレートを計算するとそれぞれ 969, 1007, 1009, 1014, 990, 1007 であり有意な差は見られなかった。

4.　おわりに

棋譜データに沿った Minimax 探索の最適化に用いられる関数 $J(\boldsymbol{v})$の形状を、連続性、偏微分不可能性、極小点数に関して解析した。2 節では損失関数等が連続で微分可能な理想的な場合を考えて、目的関数 $J(\boldsymbol{v})$は連続で、偏微分不可能な点では探索の最善応手系列が複数存在することを示した。3 節では将棋プログラムと棋譜データを用いて、現実的な場合における関数を数値解析した。ゲーム木探索が厳密には minimax 探索ではない等、理想的な状況とは異なる条件もあるが、連続性や偏微分不可能点の性質が理想的な場合と大きく異ならないことを示した。また、今回の実験条件では、極小点は複数存在するが、各局所解に対応する局面評価関数の性能は大体どれも同じであるという結果が得られた。

[1] 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.

[2] 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.

[3] 築地毅、柴原一友、但馬康宏、小谷善行、TD(λ) と Bonanza の学習法との性能比較、第 12 回ゲームプログラミングワークショップ、pp. 140–143, 2007.

[4] 築地毅、柴原一友、但馬康広、小谷善行、重ね合わせによるデータ構造を用いた評価関数の学習、第 13 回ゲームプログラミングワークショップ、pp. 144–151, 2008.

[5] 金子知適、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.

[6] 矢野友貴、三輪誠、横山大作、近山隆、既存評価関数のパラメタを活かした適応学習、第 14 回ゲームプログラミングワークショップ、pp. 1–8, 2009.

[7] G. Tesauro, Comparison training of chess evaluation functions, J. Furnkranz, M. Kumbat (Eds.), Machines that Learn to Play Games, Nova Science Publishers, pp. 117–130, 2001.

[8] Y. Tsuruoka, D. Yokoyama, T. Chikayama, Game-tree search algorithm based on realization probability, ICGA Journal, **25**, pp. 145–152, 2002

[9] T. Romstad, An introduction to late move reductions, http://www.glaurungchess.com/lmr.html.

[10] 保木邦仁、コンピュータ将棋における全幅探索と futility pruning の応用、情報処理学会誌、**47**, pp. 884–892, 2006.

[11] 伊藤裕、橋本剛、橋本隼一、動的なマージンを用いる Futility Pruning、第１２回ゲームプログラミングワークショップ、pp. 1–8, 2007.

[12] 保木邦仁、小幡拓弥、杉山卓弥、伊藤毅志、Bonanza Feliz 0.0, http://www.geocities.jp/bonanza_shogi/