

チェスの棋譜を利用した評価関数の学習

山本一成[†] 竹内聖悟[†]
金子知適[†] 保木邦仁^{††}

ゲームプログラミングの分野において、評価関数を機械学習で調整することは重要な研究課題である。なぜなら、人手での評価関数の調整は非常に手間がかかるだけでなく、多くのパラメータを矛盾なく調整するための熟練した技能が必要となるからである。一方、機械学習を用いるとゲームに関する特別に深い知識がなくとも、評価関数を作成できる。コンピュータ将棋の分野で、ボナンザメソッドと呼ばれる機械学習の手法が、実用的な評価関数の調整に初めて成功した。この手法が、ゲームプログラミング一般に通じる手法なのかは現在のところ不明である。そこで本稿では、コンピュータチェスでもボナンザメソッドが有効に働くか調査し、その効果を報告する。

Learning Evaluation Functions from Game Records in Chess

ISSEI YAMAMOTO,[†] SHOGO TAKEUCHI,[†] TOMOYUKI KANEKO[†]
and KUNIHITO HOKI^{††}

It is an important research topic in game programming to adjust evaluation functions by machine learning. Adjusting evaluation functions by hand requires not only a great deal of time, but also an expert skill to keep consistency of a large number of parameters. On the other hand, tuning evaluation functions by machine learning does not require any game-specific knowledge of human experts. In computer Shogi, a machine learning technique, “Bonanza Method”, had successfully adjusted evaluation functions for practical use for the first time. However, we are still unsure whether this method is useful in adjusting evaluation functions of various kinds of games. In this paper, we examined the performance of “Bonanza Method”, and presented some effects in Western chess.

1. はじめに

ゲームプログラミング一般において、人手での評価関数の調整は非常に手間がかかり、多くのパラメータを矛盾なく調整するのは極めて困難である。そのためゲームプログラミングの分野において、評価関数の自動調整は大きな研究課題といえる。近年、コンピュータ将棋の分野において、保木が提案した機械学習の手法によって評価関数の自動調整が可能となった。2010年現在多くのトッププログラムはこの方法、もしくはそれを改良した方法が採用されており、この手法は大きな成功を収めていると言える。しかしながらこの手法が他のゲームにも使える手法かどうかは現在のところ

不明である。コンピュータチェスはすでに多くの先行研究が存在しており、そのため手法の評価に適している。そこでチェスを題材にして、この手法が利用可能なのが調査を行った。

2. 関連研究

保木が提案した手法¹⁾との比較のために、ゲームプログラミングにおける評価関数の学習の関連研究をいくつか挙げる。なお、本稿では文献¹⁾の手法をボナンザメソッドと呼ぶ。オセロの場合では探索の結果を教師値として最適化を行う方法²⁾がある。またバックギャモンの場合ではTD-Gammon³⁾が大きな成功を収めている。チェスの場合、TD-Leaf⁴⁾、Boot-strapping⁵⁾、Tesauroが示した手法⁶⁾などが知られているが、いずれの学習方法も十分成功したとは言えない。Tesauroが示した手法⁶⁾は兄弟ノードの比較を行って学習するという点ではボナンザメソッドと同じである。しかし、Tesauroが示した手法での損失関数には拘束条件や正則化がなく、さらに重みベクトルの更新方法も異なる。重みベクトルの更新方法に関して、Tesauroの手

[†] 東京大学大学院総合文化研究科
Department of General Systems Studies, Graduate School of
Arts and Sciences, The University of Tokyo
{issei,takeuchi,kaneko}@graco.c.u-tokyo.ac.jp

^{††} 電気通信大学
University of Electro-Communications, Center for Frontier
Science and Engineering
hoki@cs.uec.ac.jp

法は最急降下法を用いていたが、ボナンザメソッドでは評価値が整数型で実装されたミニマックス探索の特性に基づき、値が数値的に収束するまで最適化を行う。なお、ボナンザメソッドでの重みベクトルの更新方法の手法の詳細は3.3章で示す。本稿ではボナンザメソッドをチェスに適用させ、この手法がチェスで利用可能かどうかを調査する。

3. 学習の枠組み

ボナンザメソッドによる学習の枠組みを簡単に説明する。なお詳細については文献^{(1),(7)}を参照されてたし。

3.1 評価関数の枠組み

チェスや将棋における、単純化した評価関数は式(1)のようになる。 w は重みベクトル、 x は局面に対応する特徴ベクトルを表す。局面の評価はこの二つのベクトルの内積として表現される。説明の簡略化のため省略するが、進行度を用いた評価関数を学習した例も存在する⁽⁷⁾ことを考えると、式(1)以外の形の評価関数でも学習可能と考えられる。

$$Evaluation = w \cdot x \quad (1)$$

3.2 損失関数

ボナンザメソッドとは、プログラムが探索を行って選んだ指し手とグランドマスターの指し手を一致させるように、評価関数の重みベクトル w を調整することである。

具体的には、グランドマスターが指した手よりも良いと評価した手の数を損失と考え、その損失を最小化する。損失は式(2)のように定式化される。なお、式(2)を以降は損失関数と呼ぶ。 $best$ はプロが指した手、すなわち教師例である。 $\xi(i)$ は i を指したあとにミニマックス探索を行った最善応手手順の末端局面の評価値となる。 T は図1にあるシグモイド関数とする。

$$Loss = \sum_{\substack{i \in \text{legal move} \\ i \neq \text{best move}}} T(\xi(i) - \xi(best)) \quad (2)$$

3.3 最適化

損失関数、式(2)を最小化するための手順を Algorithm 1 に掲載した。ボナンザメソッドでは最適化で通常使われる最急降下法もしくは擬似ニュートン法を使わない。そのかわり w の各要素に対して勾配を下る方向に一定幅動かすことを何度も繰り返し最適化を行い、さらに一定回数重みを変更したら、最善応手手順を再び作りなおして最適化を行う。こういった最適化を行う理由は損失関数は連続であるが大域的には微分可能でないからである。微分可能でない理由は、損失関数がミニマックス値を含んでおり、 w が変化すると探索結果が変わる可能性があるからである。

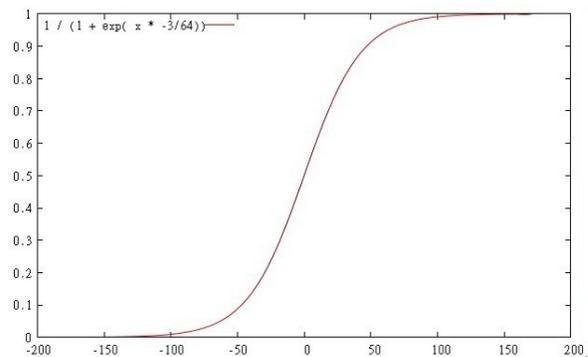


図 1 シグモイド関数

Algorithm 1 ボナンザメソッドの流れ

- (1) 棋譜のすべての合法手に対してミニマックス探索を行い、そのすべての最善応手系列を保存。
- (2) 求めた最善応手系列を使って、損失関数の勾配を下る方向に w を一定の幅更新する。同じ最善応手系列と新しい w を用いてこの更新を 16 ステップ繰り返す。
- (3) 十分最適化が進んだ場合終了、そうでないなら (1) に戻り、再び探索を行う。これを一回の反復とする。

4. 実験

本稿ではボナンザメソッドを用いて、コンピュータチェスプレイヤーの評価関数の自動調整を試みる。本実験の狙いはボナンザメソッドがチェスにおいて利用可能であるかの調査である。実験として、まず駒割の学習を試みた。駒割の学習を実験として行う理由は、実験の成否が簡単にわかるからである。さらに King と他の駒の位置評価を加えた実験について行い、学習によって棋譜との一致が進むのかどうかを調査した。

4.1 チェスの駒割

実験の前に、チェスの駒割について調べる。グランドマスターによる分析⁽⁸⁾、そして複数の代表的なコンピュータチェスの実装 (Crafty, Fruit, Glaurung) での駒割を表1に掲載した。なお比較しやすいように Rook を 500 に合わせた。駒の価値は Pawn < Knight \approx Bishop < Rook < Queen の順番であることがわかる。そして各駒の駒割の比率についても、似たような傾向が見られる。

4.2 駒割の学習の実験結果

チェスプログラム Fruit にボナンザメソッドを組み込んで、実験を行った。探索は深さ 1 の通常探索と静止探索で

<http://www.craftychess.com/>
<http://www.fruitichess.com/>
<http://www.glaurungchess.com/>

表 1 チェスの駒割 (Rook を 500 に統一)

	Pawn	Knight	Bishop	Rook	Queen
The Evaluation of Material Imbalances	100	325	325	500	925
Crafty	100	325	325	500	1050
Fruit	80/90	325	325	500	1000
Glaurung	79/100	324	324	500	996

行った．また更新幅は微分値の多い順に+2,+1,0,-1,-2と動かした．これにより駒割の総和が必ず一定になるという効果が生まれる．なお，駒割の総和を一定にする理由は，駒割の総和が定数倍変わった別解に興味がないからである．この重みの動かし方は Bonanza で行われている駒割の学習方法を参考にした．なお，実験で使用した棋譜は the International Correspondence Chess Federation の 2005 年に行われた棋譜から無作為に 1,000 局を 4 セット選んだ．4 セット別々の棋譜で学習を行うことで，本手法の棋譜に対する依存性の調査を行うのが狙いである．

図 2 は学習の経過を表している．すべての駒の価値を一定 (= 440) として学習を開始しても，学習を進めていけば，表 1 で示されているほぼ正しい駒の価値の順序になることが確認できた．また，どの棋譜を用いても駒の価値の差は最大 15 程度に収まったことを考えると，駒割の学習には棋譜は 1,000 局程度あれば十分であると考えられる．

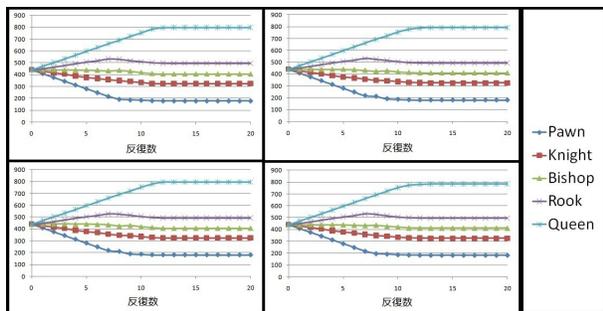


図 2 駒の価値が一定で学習を始めた場合

続いて駒割に関しての初期値の依存性を調査する．初期値として Fruit で使われてる値を使用して学習を行った．なお，他の学習条件は前の実験と同じ条件で行い，棋譜セットは 4 つあるうちの 1 つを使用した．学習結果を図 3 に載せる．Fruit で使われている値で学習のほうが駒割一定で学習を始めた場合よりも，Fruit にもともと使われている駒割に近い値となっている．以上の実験より初期値による依存性が見られる．

さらに学習した駒割と一致率について調べた．一致率とは，ある局面において棋譜の指し手が他の合法手よりも単

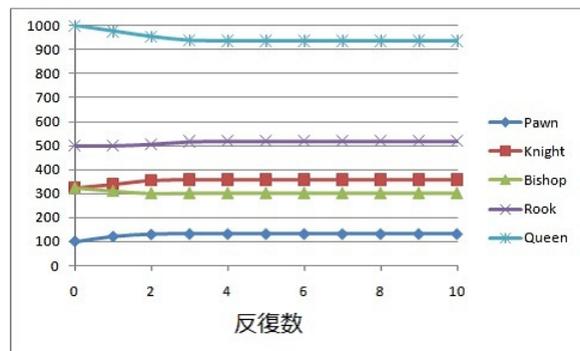


図 3 Fruit で使われている値で学習を始めた場合

独で良い評価値を出した割合と定義する．この実験で一致率の算出に使われている局面は学習に使用した局面とした．「Fruit の駒割」は Fruit で実際に使われている駒割，「学習 (Fruit)」は駒の価値の初期値を Fruit の駒割で学習させた駒割，「学習 (一定)」は駒の価値の初期値を一定とした時に学習させた駒割である．一致率は「学習 (Fruit)」がもっとも良いことから，駒の価値の初期値を一定とした時に学習させた駒割は，局所最適解の一つに収束したと考えられる．以上より，数値解の初期値依存性が確認されて，最適解に近い初期条件から始めたほうが収束が早く，より一致率の高い学習結果になると考えられる．

表 2 駒割と一致率

	Fruit の駒割	学習 (Fruit)	学習 (一定)
一致率	12.19%	14.53%	12.49%
Pawn	100	132	178
Knight	325	359	324
Bishop	325	303	404
Rook	500	520	496
Queen	1000	936	798

駒割に関する全て実験の結果からは以下のことが考えられる．学習した結果の駒の価値の割合は妥当な結果となっている．しかし手調整の駒割に比べ，Pawn の価値が高く，Queen の価値が低いのが目につく．将棋においても，学習した結果の駒割は，手調整された駒割と比べ，もっとも価値の低い駒 (チェスの場合「Pawn」，将棋の場合「歩」) の価値が高めに評価して，それ以外の駒の価値が低めに評価する傾向があり，この現象はポナンザメソッド全般の傾向と言える可能性がある．

ポナンザメソッドを行った将棋プログラム Bonanza Feliz 0.0 の駒割を参考にした

手調整の将棋プログラム YSS 7.0 の駒割を参考にした

<http://www32.ocn.ne.jp/~yss/book.html#SEC3>

<http://www.iccf.com/content/index.php>

4.3 King と他の駒の位置評価の学習の実験

ボナンザメソッドを使うことで駒割についての学習が可能であることがわかった。そこで簡単な位置評価を特徴に加えた実験を行った。評価項目としては King と他の種類の駒、2 駒の関係について約 40,000 項目を特徴に加えた。なお駒割についてはすべての駒価値が一定として学習を開始した。実験条件は 4.2 章の実験とほぼ同じであるが、過学習を回避するため、評価関数の位置評価の重みベクトルに関して L2 正則化をパラメータ 0.01 で加えた。また位置評価については、1 局面で考慮する特徴が多いことから、 w の更新幅は 0 から $\frac{3}{32}$ までの一様乱数と少なめに設定した。この重みの動かし方は Bonanza で行われている位置評価の学習方法を参考にした。駒割の重みの更新方法については 4.2 章の実験と同じである。実験に使用した棋譜は 4.2 章の実験と同じ 4 セットの棋譜集を使った。図 4 に学習の経過と一致率の関係について載せる。この場合の一致率は学習している 1,000 局の棋譜との一致率を示している。学習が進むにしたがって、徐々に一致率が上昇していることが分かる。また、どの棋譜セットを用いても一致率は 35% 付近まで上昇することから、棋譜による学習結果の差は少ないと考えられる。

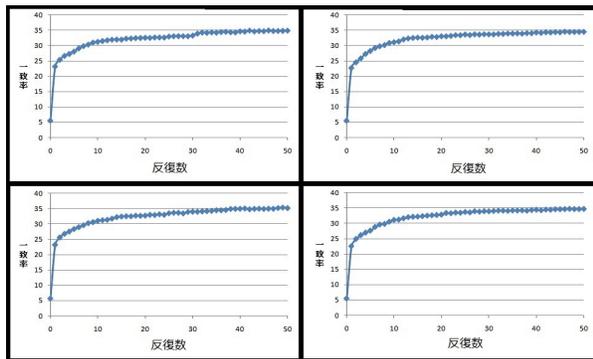


図 4 King と他の駒の位置評価の学習の経過

さらに異なるテスト用の 1,000 局面を用いて一致率を測ったところ、一致率はそれぞれ 32.92%、33.43%、33.13%、35.44%となった。これは Fruit の元の評価関数を使った場合の一致率 32.18%を若干ながら上回る結果である。以上の結果からボナンザメソッドがチェスにおいて利用可能であると考えられる。各駒の移動可能範囲や、ピンされているかどうか、さらに序盤や終盤によって評価関数を変える、といったチェスプログラムで一般的に使われている特徴を加えることで、さらに一致率も上昇すると予想される。なお学習における計算時間の大半は最善応手手順の末端局面を求めることに費やされるため¹⁾、学習する特徴が増加した場合でも計算コストの増大は許容できる範囲だと考えられ

る。実際に、将棋プログラム Bonanza は 5000 万以上ものパラメタの学習に成功している。

5. おわりに

将棋におけるボナンザメソッドの有効性は、すでに多くのトッププログラムが示している。現在は多くの強豪将棋プログラムの評価関数は、手調整ではなくボナンザメソッドもしくはそれを改良した教師あり学習に切り替わっている。この事実は人手での調整よりも、学習が算出した評価関数のほうがより良いためと考えられる。

本稿ではチェスにおいても、ボナンザメソッドが有効に働く可能性を示し、また駒の価値と一部の簡単な位置評価については学習することに成功した。今後、様々な特徴ベクトルを評価関数に組み込み、学習を行うことで、チェスにおいても将棋と同様に実用に耐える評価関数の作成が可能となると期待できる。

参考文献

- 1) 保木邦仁.(2006). 局面評価の学習を目指した探索結果の最適制御. *Proceedings of 11th Game Programming Workshop*, pp.78-83.
- 2) M. Buro. (2002). Improving Heuristic Mini-max Search by Supervised Learning. *Artificial Intelligence*, 134(1-2), pp.85-99.
- 3) Tesauro, G.J. (1992). Practical Issues in Temporal Difference Learning, *Machine Learning*, Vol.8, pp.257-277.
- 4) J.Baxter, A. Tridgell, and L. Weaver. (2000). Learning to Play Chess Using Temporal Differences. *Machine Learning*, 40(3) pp.242-263.
- 5) J. Veness, D. Silver, W. Uther, A. Blair. (2009). Bootstrapping from Game Tree Search, *Advances in Neural Information Processing Systems 22*, pp.1937-1945.
- 6) Tesauro, G.J. (2001). Comparison training of chess evaluation functions *Machines that learn to play games*, pp.117-130
- 7) 金子知適.(2008). 将棋の棋譜を利用した、大規模な評価関数の調整. *Proceedings of 13th Game Programming Workshop*, pp.152-159.
- 8) Larry Kaufman. (1999). The Evaluation of Material Imbalances. *ChessLife, March*