情報処理学会論文誌　Vol. 51　No. 11　2031–2039 (Nov. 2010)

囲碁における勾配法を用いた確率関数の学習

松井利樹[†1]　野口陽来[†2]
土井祐紀[†2]　橋本剛[†3]

囲碁の指し手を評価するには膨大な数の特徴とそれらの複雑な関係を考慮する必要がある．人手によるパラメータの調整はほぼ不可能であり機械学習が唯一の現実的なアプローチである．本稿ではコンピュータ将棋において評価関数の学習に用いられている勾配法をコンピュータ囲碁に応用する．将棋の評価関数と囲碁の評価関数では求められるものが異なる．将棋では手が正しく順序付けられれば十分であるが，囲碁ではモンテカルロシミュレーションの確率分布を生成するため，比率も適切でなくてはならない．本稿では異なる 2 つの誤差関数を設計することでこの問題を解決している．ベンチマークとして Bradley-Terry モデルと Elo レーティングモデルを用いた学習手法（これは世界最強の囲碁プログラムの 1 つ Crazy Stone で用いられている）と比較した結果，大きな性能向上を確認できた．

A Gradient Method for the Evaluation Function in the Game of Go

Toshiki Matsui,[†1] Haruki Noguchi,[†2] Yuki Doi[†2]
and Tuyoshi Hashimoto[†3]

To evaluate moves in the game of Go, a large number of features and their complicated relationships have to be considered. These features are nearly impossible to be optimized by hand so machine learning is one method. We apply a gradient method which is used in computer Shogi for learning of the evaluation function. Evaluation function for Shogi and Go have different characteristics. In Shogi, it is enough to be able to order moves correctly for use in alphabeta. While in Go, moves require proper score in order to generate a probability distribution for use in Monte-Carlo simulation. We solve this problem by designing two error functions. We compare our method to Bradley-Terry and Elo-Rating model that are used in Crazy Stone, which is one of the best program in the world. Experimental results show that our method produces a stronger Go player.

1. はじめに

ゲームにおける機械学習の多くのターゲットは評価関数の自動調整にある．古くはバックギャモン[1)] やオセロ[2)] などに学習が応用され一定の成果を出している．しかし将棋や囲碁などの難しいゲームでは複雑な局面をうまく学習することは困難であった．そのため大部分のゲームで評価関数はプログラマの手で設計するのがつねであった．そのような状況下で 2005 年に世界コンピュータ将棋選手権において，局面評価関数を強化学習によって調整された Bonanza が優勝し，将棋の強化学習の走りとなった[3)]．またコンピュータ囲碁でも『Computing Elo Ratings of Move Patterns in the Game of Go』[4)] によって Bradley-Terry モデルと Elo レーティングの概念を用いた学習手法（以後，Elo レーティングモデルと呼ぶ）が考案され，囲碁における確率関数の自動調整を実現している．彼によって作られた Crazy Stone は現在世界最強の囲碁プログラムの 1 つとなっている．彼らの活躍によって将棋や囲碁などの難しいゲームにおいても強化学習がきわめて有効であることが示された．モンテカルロ囲碁の誕生以来，囲碁は現在大きな盛り上がりを見せており，今後のさらなる躍進の原動力として大きな期待をされているのが機械学習である．囲碁ではゲームの特性上，確率関数の特徴数が膨大になりがちなため手作りによる調整は困難である．そのため機械学習によるアプローチが考えられるが，優れた学習を実現することは難しい．囲碁の機械学習で大きな成果をあげたのは前述した Elo レーティングモデルであり，本稿では将棋で成功した勾配法によるアプローチを使って Elo レーティングモデルを上回る学習システムの構築を目指す．

2. 関連研究

モンテカルロ囲碁における確率関数の学習の最も単純なアプローチは，ノビやトリなどの各特徴が選ばれる頻度を棋譜から測る方法である[5)]．ある特徴が打たれた回数をその出現回数で割り評価値とする手法である．この手法の大きな弱点は着手以外の手を学習に生かせ

†1 株式会社 KDDI 研究所
KDDI R&D Laboratories Inc.
†2 北陸先端科学技術大学院大学情報科学研究科
School of Information Science, Japan Advanced Institute of Science and Technology
†3 松江工業高等専門学校情報工学科
Department of Information Engineering, Matsue College of Technology

ないので各特徴の強さをうまく調整できない点にある．Stern らは，着手以外の手を学習に取り入れる問題を扱っている[6]．このモデルでは各特徴の強さを精度良く計算できる．しかしこのアプローチの弱点は，評価に必要な数が特徴の個数につれて指数関数的に増えるため，特徴の個数に大きな制約がかかることにある．そこで登場したのが，Bradley-Terry モデル[7] と Elo レーティング[8] のアルゴリズムを組み合わせた手法である[4]．この手法は着手以外の手を学習に組み込むことができ，特徴の数に対して計算コスト，使用するメモリ量は健全であり，囲碁のような大規模な非線形計画問題に対して有効で，現在のコンピュータ囲碁に広く用いられている．また，コンピュータ将棋では保木による強いプレイヤと同一の指し手を選択する局面評価関数の発見を目指し，勾配法を用いて学習を行う手法が提案された[3]．この手法はコンピュータ将棋に大きな影響を与え，その強さを大きく向上させた．

2.1 Bradley-Terry モデルと Elo レーティングモデル

Bradley-Terry モデルは勝敗を予測するモデルである．あるプレイヤ i（$1 \leq i \leq n$）が γ_i という正の値（レーティング）を持つとすると，n 人のプレイヤの中で i が勝利する確率は，下記のように表せる．

$$P(i) = \frac{\gamma_i}{\sum_{j=1}^{n} \gamma_i} \tag{1}$$

複数プレイヤによるチームでの試合は，チームの強さをチームを組むメンバの強さ γ_i の積とすることで，扱うことができる．プレイヤ 1，2，3，4，5，6 が存在し，1-2-3，4-5，6 という 3 つのチームを組むとき，チーム 4-5 が試合に勝つ確率は，下記のように表せる．

$$P(\text{4-5}) = \frac{\gamma_4\gamma_5}{\gamma_1\gamma_2\gamma_3 + \gamma_4\gamma_5 + \gamma_6} \tag{2}$$

一般的には，γ_i を $400\log_{10}\gamma_i$ と変換したものを Elo レーティングと呼ぶ．この Elo レーティングを minorization-maximization アルゴリズムを使って最適な値を求めることで学習を実現している．

2.2 保木の手法

保木の手法では，以下の目的関数 $J_x(S)$ を最小化するパラメータベクトル x を反復学習により求める．誤差汎化関数 $T(x)$ には適当なスケーリングを施したシグモイド関数を用いる．

$$J_x(S) = \sum_{s=1}^{S}\sum_{m=1}^{M} T[f_x(p_{s,m}^{leaf}) - f_x(p_{s,0}^{leaf})] + \Phi_x \tag{3}$$

ここで，S はサンプル局面の集合であり，M はサンプル局面 p_s から選択できる手の集合である．ある局面 p_s から選択された手 m によって遷移した局面 $p_{s,m}$ から探索を行い，評価関数 f_x によって推定された最善局面のスコア $f_x(p_{s,m}^{leaf})$ を訓練データとし，十分に強いプレイヤによって選択された手（$m = 0$）から探索を行い評価関数 f_x によって推定された最善局面のスコア $f_x(p_{s,m}^{leaf})$ を教師信号とする．また過学習を避けるために駒の価値に基づいた拘束条件 Φ_x を付与することで，学習を安定化している．この数式に関する詳しい説明は 3.2.1 項で行う．

3. 既存手法を用いた学習の設計

本研究では，将棋における局面評価関数の学習として設計された保木の手法をモンテカルロ囲碁における局面遷移確率の学習に適用し，Elo レーティングを超える性能を持った機械学習アルゴリズムの構築を目指す．保木の手法は，将棋の局面評価関数の学習を目指した設計となっており，モンテカルロ囲碁で適用する場合は課題がいくつか残る．4 章ではそれらの課題を整理し，課題の解決方法を提案する．

3.1 Bradley-Terry モデルによる確率関数の設計

モンテカルロ囲碁では局面評価にモンテカルロ法を利用するため，局面評価関数は必要ない．代わりに評価関数は手を評価するために用いられる．ここでは文献 9) と同様に，手の評価値から木探索中での枝刈りやプレイアウト中での確率分布の生成を行う．局面評価関数は線形和で与えられることが多いが，モンテカルロ囲碁では確率を表現するため積の形がよく採用される．確率を表現するうえで積の方が優れている要因として評価値が非負であることと，加減算と比べて乗算の方が評価の強弱をつけやすい点があげられる．ある手 p が与えられたときの評価関数 $\gamma(p)$ は以下のように定義される．

$$\gamma_x(p) = \prod_{k}^{K} l_k(p) \tag{4}$$

ある手 p が与えられたとき，特徴があるかないか判定する特徴関数 $l_k(p)$ は以下のように定義される．

$$l_k(p) := \begin{cases} x_k & (\text{特徴がある}) \\ 1 & (\text{特徴がない}) \end{cases} \tag{5}$$

パラメータベクトルである $\boldsymbol{x}$ は以下のように表記される．

$$\boldsymbol{x} := [x_1, \cdots, x_K]^T \quad (x_n > 0,\ n \in K) \tag{6}$$

ある局面からある手 p_i が打たれる確率を求める確率関数 $f_x(p_i)$ は Bradley-Terry モデルにならって，以下のように定義される．

$$f_x(p_i) = \frac{\gamma_x(p_i)}{\sum_{m=0}^{M} \gamma_x(p_m)} \tag{7}$$

3.2　保木の手法を囲碁に適用

ここでは保木の手法を囲碁に適用し，勾配法を使った教師あり学習の設計を行う．多くのゲームで十分強いプレイヤの棋譜を集めることは容易ではないが，囲碁ではサンプル数は十分に確保できる．

3.2.1　誤 差 関 数

保木の手法と同じく本手法でも棋譜中で選択された手はその局面で選ばれなかったどの手よりも価値が高いとして，棋譜中で選択された手の確率を最大化することを考える．棋譜サンプル中のある局面 p_s でプレイヤが選択した手 $p_{s,0}$ を教師信号とし，教師信号以外のすべての手 $p_{s,m}$ を訓練集合として，誤差関数 $J_x(S)$ は以下のように定義する．

$$J_x(S) = \sum_{s=1}^{S} \sum_{m=1}^{M} T[f_x(p_{s,m}) - f_x(p_{s,0})] + \Phi_x \tag{8}$$

$$T[x] = sig(x) = \frac{1}{1 + e^{-\eta x}} \tag{9}$$

S はサンプル局面の集合であり，M はサンプル局面 p_s から選択できる手の集合である．保木の手法と異なる点として，評価関数 $f_x(p)$ に局面評価関数を使うのではなく Bradley-Terry モデルにならって定義された確率関数を使うこと，局面の探索を行わないこと，拘束条件式が異なることがあげられる．誤差を定量化する誤差汎化関数 $T[\cdot]$ については保木の手法と同じくシグモイド関数を用いる．拘束条件 Φ_x については 4.2 節で説明する．

4.　既存手法の課題と対策

棋譜からは「強いプレイヤが選択した手」の情報が得られるため，棋譜で選択された手を教師信号と見なし，その他の手を教師信号よりも小さくするという学習が行われる．これにより，教師信号の評価を最大化する確率関数の発見が可能になる．しかし，優れた確率関数の設計には教師信号の評価を最大化するだけでは十分でなく，各特徴の強さを精度良く計算する必要がある．Elo レーティングモデルはレーティングの概念により，各特徴の強さを評価している．また保木の手法は，将棋において重要な特徴である駒の価値を利用して各特徴の強さを評価している．しかし，囲碁では将棋における駒の価値のような絶対的価値を持つ特徴は知られておらず，保木の手法を使って特徴の強さを調整することは困難である．本稿ではより汎用的な方法で特徴の強さを調整する手法を 4.1 節で提案する．また，パラメータへ付与する拘束条件を 4.2 節で提案し，パラメータの更新方法も 4.3 節で提案する．これらを実現することにより優れた学習の実現を目指す．

4.1　パラメータの調整

どのような指標で特徴の強さを調整するかが最も重要な要素となる．様々な実験を重ねた結果，確率関数によって n 位と判断された手が棋譜上の手に一致する割合（**一致率**）を基準に特徴の強さを調整すると，高いレーティングを得ることができたため，一致率を基準としてパラメータの調整を行う方針をとる．一致率を基準に調整を行うことで，棋譜との一致度が高い場合は各手に大きなバイアスを与え，逆に棋譜との一致度が悪い場合は，各手の確率は一様に近くなる．これは予測性能から各手に与えるバイアスを制御しようとする試みであり，人間の感覚的にも妥当なアプローチであると思われる．

4.1.1　誤 差 関 数

確率関数 f を局面 s のすべての指し手に適用したときに，その大小関係で決定される指し手 m の順位を $rank_f(s,m)$ と表す．この順位関数を学習対象のすべての局面に適用し，順位 r の手が実際に選択される割合である一致率 $\chi(r)$ を，次のように定義できる．

$$\chi(r) = \frac{|\{m_{s,0} \mid s \in S,\ rank_f(s, m_{s,0}) = r\}|}{|S|} \tag{10}$$

ここで $m_{s,0}$ は局面 s で選択された手である．一方，f によって実際に順位 r と判断される手の平均的な確率 $\chi'(r)$（**平均確率**）は $\chi(r)$ とは異なり，次のように定義できる．

$$\chi'(r) = \frac{\sum_{s \in S} f(p_{s,m_r})}{|S|} \tag{11}$$

ただし m_r は $rank_f(s, m_r) = r$ となる指し手．$\chi'(r)$ を $\chi(r)$ に近づけるために f_x をフィルタリングした関数 f_z を導入して誤差関数を次のように定義した．

$$L_z(S) = \sum_{s=1}^{S} \sum_{m=1}^{M} [f_{\boldsymbol{z}}(p_{s,m}) - \chi(rank_{f_x}(s,m))]^2 \tag{12}$$

$$\boldsymbol{z} := [z_1, \cdots, z_K]^T \quad (z_n > 0,\ n \in K) \tag{13}$$

ここで $\boldsymbol{z}$ はパラメータベクトル $\boldsymbol{x}$ をフィルタリングしたものである．これは，たとえば $\chi(1)$ の手が棋譜で選択される確率が 35%のとき，$\chi(1)$ の手の確率が平均的に 35%であることを要求するという意味である．誤差汎化関数には自乗誤差を使用する．

4.1.2　フィルタの設計

パラメータベクトル $\boldsymbol{x}$ を $\boldsymbol{z}$ へ変換するためにフィルタを設計する必要があるが，このとき，フィルタの特性として以下の性質を満たさなければならない．

- $\chi(r)$ が $\boldsymbol{z}$，$\boldsymbol{x}$ とも変わらない（順位が保証される）
- 値が正のままである（確率の生成に支障を与えない）

これを実現するには単純にパラメータの各要素を累乗すればよい．評価関数 $\gamma(p)$ は積であるから，上記の条件を満たすことができる．よって $\boldsymbol{z}$ は以下のように定義される．

$$z_k = (x_k)^{\xi} \tag{14}$$

結局，係数である ξ を調整することにより，学習を実現できる．

4.2　パラメータの拘束条件

本手法でも保木の手法と同様に学習を安定させるためにパラメータへ拘束条件を付与することを考える．保木の手法は線形和で表現された評価関数に対して付与するものであることや，将棋のゲーム特性を考慮して設計されていることもあり，今回の学習に利用することができない．そこで，本手法に適した新たな拘束条件を設計することを考える．各パラメータが必要以上に基準値となる「1」から離れないような条件をヒューリスティックに設計した．

$$\frac{d}{dx}\boldsymbol{\Phi} = [\phi_1, \cdots, \phi_k]^T \tag{15}$$

$$\phi_k = \omega \cdot \left(x_k - \frac{1}{x_k}\right) \cdot \rho(k) \tag{16}$$

出現頻度の高いものは勾配の絶対値が大きいために拘束の影響を受けにくくなるため，サンプルごとの出現回数を考慮して，拘束を与えた．w はペナルティの強さを決める定数であり，$\rho(k)$ は全サンプル中 x_k が出現した回数を返す関数である．拘束条件を付加すると棋譜との一致率は減少してしまうが，その代わり学習が安定する．

4.3　パラメータの更新

勾配ベクトルを計算し，勾配方向に従って反復的にパラメータベクトルを更新していくことで値が収束する．確率関数はパラメータベクトルの積で表現されているため，保木の手法でとられた手法を流用してもまったく収束しなかった．そこで，本手法に適した更新式を設計することを考える．文献 10) を参考に，以下のように設計した．

$$x_k^{N+1} = \begin{cases} x_k^N \cdot (1 + a_k^N) & (\partial_{x_k} J_x^N > 0) \\ x_k^N / (1 + a_k^N) & (\partial_{x_k} J_x^N < 0) \end{cases} \tag{17}$$

a の値は以下のようにステップごとに変化させていく．

$$t_k = \partial_{x_k} J_x^{N-1} \cdot \partial_{x_k} J_x^N \tag{18}$$

$$a_k^{N+1} = \begin{cases} a_k^N \cdot \beta & (t_k > 0) \\ a_k^N / \alpha & (t_k > 0) \end{cases} \tag{19}$$

$$1 < \beta < \alpha \tag{20}$$

また，特徴サンプルが非常に少ないものを学習に加えると，不安定になってしまう．そこで $\rho(k) < \lambda$ ならば，学習は行わず 1 にするようにした．λ はプログラマが与える定数である．この手法は最急降下法において特徴ごとに刻み幅を持たせたヒューリスティックな手法である．特徴は前回得られた勾配ベクトルと今回得られた勾配ベクトルの符号の変化によって刻み幅の調整を行うことにある．符合が反転したときに刻み幅を小さくし，符号が等しいときに刻み幅を大きくすることで安定して収束する．優れた収束のためには α，β を適当に設定する必要がある．

5.　特徴の設計

囲碁でよく使われる特徴は，「直前手からの距離」のようなヒストリー，「ノビ」「アタリ」のような囲碁の重要な分野知識，そして「着手点周辺の石の形」である．囲碁は直前手の近辺が最善手である確率が非常に高く，直前手からの距離は重要な特徴となっている．文献 4) を参考に，2 つの座標間の距離 d を以下の式で数値化する．

$$d(\delta x, \delta y) = |\delta x| + |\delta y| + max(|\delta x|, |\delta y|) \tag{21}$$

以下距離が必要とされる箇所ではこの式を用いる．これから設計する特徴の具体的な数値を**表 1** に掲載する．

I：着手点の座標

天元（盤の中心）と着手点との距離 *length* で場合分けする．

II：直前手からの距離

直前手と着手点の距離 *length* で場合分けして計算する．

III：二手前からの距離

二手前の手と着手点の距離 *length* で場合分けして計算する．

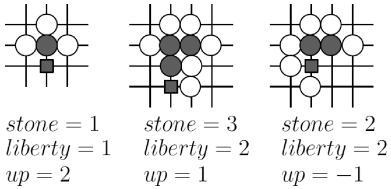
IV：ノビに関する特徴

ここでいうノビとはアタリになっている連だけを指すのではない．隣接する連の *liberty* の数が 1 または 2 のときに計算される．対象連の石の数 *stone* と着手する対象連の *liberty*

表 1　特徴ベクトルの設計
Table 1　Design of feature vector.

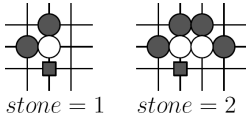
	UCT(九路)	MC(九路)
I	$length = (1, 2, \ldots, 15$ 以上$)$	$length = (1, 2, \ldots, 15$ 以上$)$
II	$length = (1, 2, \ldots, 9$ 以上$)$	$length = (1, 2, \ldots, 3$ 以上$)$
III	$length = (1, 2, \ldots, 9$ 以上$)$	-
IV	$stone = (1, 2, 3$ 以上$)$ $up = (-1, 0, \ldots, 2$ 以上$)$	$stone = (1, 2, 3$ 以上$)$ $up = (-1, 0, \ldots, 2$ 以上$)$
V	$stone = (1, 2, 3$ 以上$)$	$stone = (1, 2, 3$ 以上$)$
VI	$stone = (1, 2, 3$ 以上$)$ $up = (-1, 0, \ldots, 2$ 以上$)$	$stone = (1, 2, 3$ 以上$)$ $up = (-1, 0, \ldots, 2$ 以上$)$
VII	$stone = (1, 2, 3$ 以上$)$	$stone = (1, 2, 3$ 以上$)$
VIII	$size = 4$	$size = 3$

の増分 *up* で場合分けする．



V：アタリに関する特徴

隣接する相手の連の *liberty* の数が 2 のときに計算される．対象連の石の数 *stone* によって場合分けする．



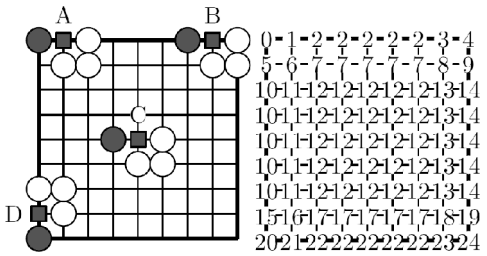
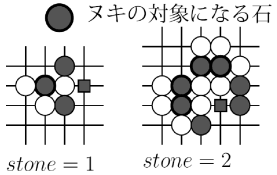


図 1　パターンと位置情報
Fig. 1　Pattern and location information.

VII：ヌキに関する特徴

隣接する相手の連の *liberty* の数が 1 のときに計算される．着手によってとることができる相手の連が自分の石を当てている場合にのみ計算される．複数の連を当てているときはそのつど計算される．当てられている自分の連の石の数 *stone* によって場合分けする．



VIII：着手点周囲の形の特徴

着手点からの距離 d が *size* 以下である範囲の石の配置を特徴に組み込む．自分の石，相手の石，空白 or 盤端という 3 種類を考慮し，位置情報を石の形に付加する．これより石の形に位置情報を与えることができるのだが，特徴の組合せが大きくなってしまうのが欠点である．そこで回転と反転を考慮することで，**図 1** の A，D のような石の形は異なるが本質的に同じ特徴を同一視する．

上記の 8 項目から特徴を設計する．木探索中で用いる確率関数（UCT）とプレイアウト中で用いる確率関数（MC）は求められる速度が違うため，それぞれ表 1 のように設計した．

6. 実験と考察

6.1 パラメータ設定

プログラマが学習システムに与えなければならない環境は以下のとおりである．

表 2　設定した定数
Table 2　Settings of parameter.

	サンプル局面数	η	ω	λ	α	β	a
九路盤	808679	1000	$1 \cdot 10^{-10}$	100	1.8	1.2	0.5

今回は九路盤での環境で実験を行い，**表 2** のように設定した．

- 特徴関数

5 章で述べた．

- 棋譜サンプル

九路盤では強い人間プレイヤの棋譜があまり存在しないので，学習に使う棋譜サンプルは十分に強いプログラム（CGOS（http://cgos.boardspace.net/）で Rating2200 以上）どうしの対局の棋譜を使用した．この Rating であれば人間プレイヤと比較しても十分強い．しかしコンピュータどうしの対局は勝負が決まっても終局まで打ち続けてしまうため，終盤の手は非常に精度が悪い．そこで棋譜サンプルは初手から 45 手までとした．

- η（sigmoidscale）
- ω（penaltyscale）
- λ（samplescale）
- α，β，a（learningscale）

6.2　実 験 環 境

本稿で提案した勾配法を使った学習と，Elo レーティングモデルによって学習した結果を以下の 2 つの観点から比較する．

- 2 つの学習手法の性能比較
- 2 つのプログラムの対戦による比較

特徴ベクトルや学習に使った棋譜サンプルは両方とも同じものを使い，学習手法以外はすべて同じ条件で実験した．

6.3　成功率と一致率の比較

図 2 より成功率（教師手以外の手が教師手よりも評価値が低いと判断できる確率）は勾配法の方がわずかに上回った．**図 3** より，一致率の比較では顕著な差がみられなかった．2 つの指標からは，両手法のはっきりとした違いを見い出すことはできなかった．

6.4　一致率と平均確率

平均確率と一致率を比較すると，**図 4** より，勾配法ではパラメータの調整前では棋譜と

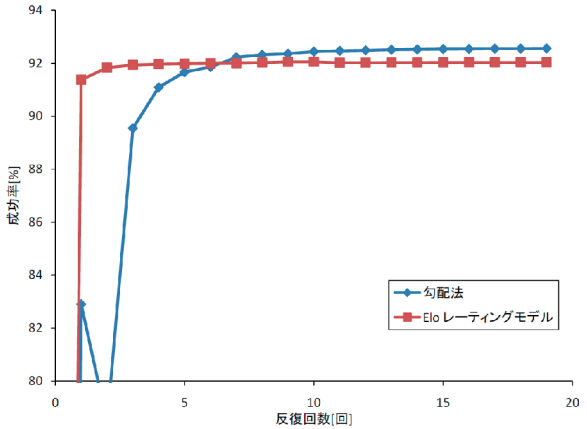


図 2　成功率の比較
Fig. 2　Comparison of success rates.

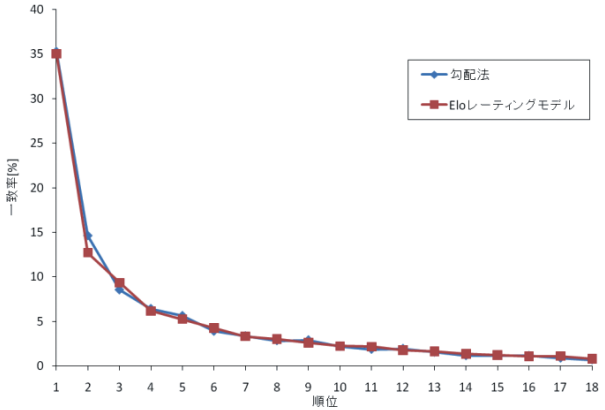


図 3　一致率の比較
Fig. 3　Comparison of concordance rates.

の一致率に大きな差があることが分かる．しかし，パラメータを調整することによって一致率と平均確率をほぼ一致させることができる．パラメータ調整前のプログラムと調整後のプログラムを自己対戦させると調整後のプログラムが 8 割近く勝利するため，棋力の向上

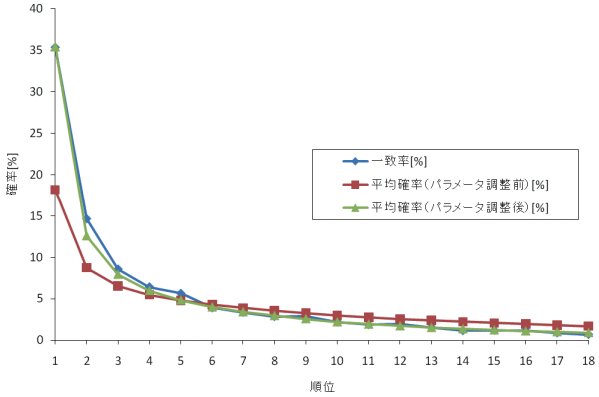


図 4　勾配法の平均確率と一致率の比較
Fig. 4　Comparison of success rates and average rates for gradient method.

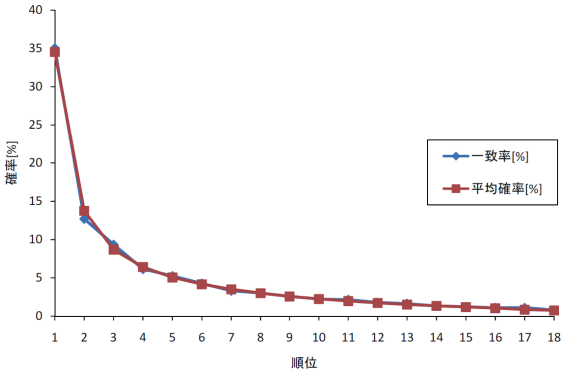


図 5　Elo レーティングモデルの平均確率と一致率の比較
Fig. 5　Comparison of success rates and average rates for Elo rating.

はあきらかである．逆に Elo レーティングでは一致率と平均確率にほとんど差がみられない（**図 5**）．一致率に平均確率を合わせることが最適であるか証明できていないが，十分に妥当性のある指標に対して両手法とも適切に値を調整することに成功している．

6.5　学習にかかる時間

1 回の反復にかかる時間は Elo レーティングモデルよりも計算が複雑なことが影響して勾配法の方が遅く，収束に必要な反復回数も勾配法の方が 2 倍程度必要になる．総じて勾配法は Elo レーティングモデルと比べて 3 倍程度の時間がかかる．

	天元 (5,5) に打つ	盤の角 (1,1) に打つ
x_{koubai}	1.80	0.40
x_{elo}	38.5	0.15

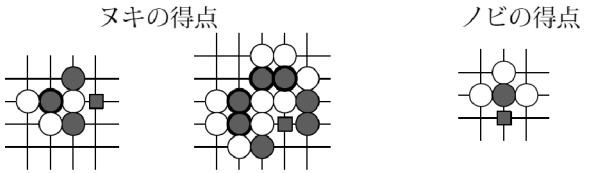


図 6　スコアの比較
Fig. 6　Comparison of score.

6.6　学習したパラメータの比較

勾配法と Elo レーティングモデルによって得られたパラメータの中で顕著な差が現れたものを比較した．囲碁のゲーム特性を考慮するとヌキは重要な特徴であり，特に盤面が小さい 9 路盤であれば中央の三子抜きはほぼ勝利に直結するであろうし，一子抜きであっても勝敗の重要な要素となることが多い．またノビの手は盤が狭い 9 路盤において有利に働くことが多い（**図 6** 左下）．逆に多くのプログラムが選択する初手天元はそれほど有利とは思えない．Elo レーティングモデルは序盤の定石などによく打たれるが，それほど有利とはならない手に対して大きな得点を与え，勝敗に直結する特徴に対して大きな得点を与えることができていない．逆に本手法は，序盤の定石などには大きな得点を与えず，勝敗に直結する特徴に対して大きな値を与えることができている．人間の感覚で考えると，序盤の定石と中盤の確定手（そこに打たなければ敗着となる絶対的な一手）は選択される確率は等しくても，手の価値で考えれば大きな差があるべきである．これは 2 つの学習手法の差から生まれてくるものであり，本手法はシグモイド関数と拘束条件の制御が働いて，必要以上に大きな値に調整しようとしない．しかし，Elo レーティングモデルではレーティングで価値を判断するため，初手天元のような多くのプログラムが選択するが大きな有利とはならない手に必要以上の大きな値を与えてしまっている．

6.7 勾配法 vs. Elo レーティングモデル

我々が開発した囲碁プログラム Nomitan に，勾配法と Elo レーティングモデルによって学習された確率関数を実装し対戦による実験を行った．勾配法，Elo レーティングモデルとも使用する特徴は同一である．交互に先後入れ替えを行い 200 回対戦を行ったところ，142 勝 58 敗と有意に勝ち越すことができた．

7. ま と め

一致率や成功率といった指標では両手法の間で明確な差が生まれなかったにもかかわらず，対戦結果で大きな差が生まれた．これは，一致率や成功率が必ずしも棋力の向上に直結しないということを示している．さらに平均確率でも勾配法と Elo レーティングモデルでは大きな変化がみられない．両手法において強さに明確な差が現れた理由として，Elo レーティングモデルでは定石などのよく選択されるが大きな有利とはならない手に必要以上に高いスコアを与えてしまい，精度良く学習できていない点があげられる．勾配法は Elo レーティングモデルと比較すると，プログラマに対して調整を要求する変数や設定が多く，収束に必要な時間も長い．しかし，Elo レーティングモデルよりもゲームの特性を精度良く学習することができ，対戦では有意に勝ち越すことができた．提案した学習法はコンピュータ囲碁におけるパラメータ学習で有望な手法であるといえるだろう．

参 考 文 献

1) Tesauro, G.: Temporal Difference Learning and TD-Gammon, *Comm. ACM*, Vol.38, No.3, pp.58–68 (1995).
2) Buro, M.: Experiments with Multi-ProbCut and a New High-Quality Evaluation Function for Othello, Games in AI Research, van den Herik, H.J. and Iida, H. (Eds.) (2000).
3) 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).
4) Coulom, R.: Computing Elo Ratings of Move Patterns in the Game of Go, *Computer Games Workshop* (2007).
5) Bouzy, B. and Chaslot, G.: Bayesian Generation and Integration of K-nearest-neighbor Patterns for 19x19 Go, *IEEE 2005 Symposium on Computational Intelligence in Games*, pp.176–181 (2005).
6) Stern, D., Herbrich, R. and Graepel, T.: Bayesian Pattern Ranking for Move Prediction in the Game of Go, *Proc. International Conference of Machine Learning*, pp.873–880 (2006).
7) Hunter, D.R.: MM Algorithms for Generalized Bradley-Terry Models, *The Annals of Statistics*, Vol.32, No.1, pp.384–406 (2004).
8) Elo, A.E.: *The Rating of Chess Players, Past & Present*, Arco Publishing (1978).
9) Gelly, S., Wang, Y., Munos, R. and Teytaud, O.: Modification of UCT with Patterns in Monte-Carlo Go, Technical Report RR-6062, INRIA (2006).
10) 松井利樹，橋本　剛，橋本隼一：勾配法を使った学習の収束に関する研究，第 13 回ゲームプログラミングワークショップ，pp.92–95 (2008).

(平成 22 年 1 月 25 日受付)
(平成 22 年 9 月 17 日採録)

松井　利樹

1984 年京都生まれ．2007 年京都工芸繊維大学機械システム工学科卒業．2009 年北陸先端科学技術大学院大学情報科学研究科博士前期課程修了．2009 年（株）KDDI 研究所入社．コンピュータ将棋の開発，コンピュータ囲碁の開発，機械学習の研究のほか，近年ではモバイルセキュリティ，ソフトウェアの著作権保護技術に従事．

野口　陽来

1984 年富山生まれ．2007 年富山高等工業専門学校専攻科機械・電気システム工学専攻修了．2009 年北陸先端科学技術大学院大学情報科学研究科博士前期課程修了．コンピュータ将棋の開発，コンピュータ囲碁の開発に従事．

土井　佑紀

2010 年北陸先端科学技術大学院大学情報科学研究科博士前期課程 2 年．現在，同大学にてコンピュータ囲碁プログラムの作成に参加．

橋本　　剛

1970 年京都市生まれ．1998 年静岡大学にてコンピュータ将棋を中心にゲーム情報学の研究に従事．2001 年静岡大学工学博士．学術振興会特別研究員，北陸先端科学技術大学院大学講師を経て現在松江工業高等専門学校情報工学科准教授．コンピュータ将棋 TACOS の開発，証明数探索の研究のほか，近年はコンピュータ囲碁やオセロの完全解析の研究も行っている．