

集団学習を用いた合議制囲碁プレイヤー

野 中 翔 平^{†1} 中 村 貞 吾^{†1}

複数のプレイヤーにより多数決を行う合議は、囲碁でも有効であることが示されている。乱数によって生成される複数のプレイヤーからなる合議は容易に作成できるが、合議では個々人の能力よりも集団としての多様性が重要であるといわれており、乱数のみでは多様性が少ない。本研究では、他のプレイヤーの知識を考慮した集団学習を行うことにより多様性を持たせた知識を用いる。結果として、個別に学習を行った知識による合議よりも他のプレイヤーを考慮することで多様性を持たせた合議の勝率が高くなった。

Consultation Go Players using Ensemble Learning

SHOHEI NONAKA^{†1} and TEIGO NAKAMURA^{†1}

It is effective to make consultation with a number of algorithms to play Go. We can easily construct each algorithm which uses a base evaluation function plus some small random values, but these algorithms have lack of variety. Because the variety of group is more important than the ability of individuals in consultation, we propose a method for increasing the variation of group of algorithms utilizing knowledge based on ensemble learning. We present some experimental results that show our method improves the winning rate.

^{†1} 九州工業大学情報工学部

Kyushu Institute of Technology School of Computer Science and Systems Engineering

1. はじめに

2008年に初めて合議による将棋プログラムとして、5五将棋に対する合議プログラムが登場した。これは思考プログラムの違う4つの将棋プログラムに同じ局面を与え、得られた手から単純多数決によって最終的な手を決定するというものである。結果として、単体のプログラム全てに勝ち越すことで合議の有効性を示した。その後、将棋プログラムのBonanzaに対し乱数を用いることで擬似的に複数のプログラムを用意し、それらの多数決によって手を打つ文殊が合議プログラムの有効性を更に示した。このことにより合議に対する研究が盛んになり、囲碁に対しても合議が有効であることが確認されている¹⁾。

複数の思考プログラムを用意する必要のあった5五将棋の合議プログラムに対し、文殊のように単体のプログラムに対して乱数による擬似的な複数化は実装が用意であり、構造のシンプルさといった利点がある。しかし、乱数のみによるプログラムの複数化は個々のプレイヤーが似た性質を持ってしまう可能性がある。合議では個々の能力よりも集団としての多様性が重要であるといわれている。そこで、将棋プログラムの合議に対して評価関数を学習する際に、他のプレイヤーの学習結果を考慮した集団学習を行うことで多様性を増すことの有用性が示されている²⁾。

本研究では、UCTを用いたモンテカルロ法による囲碁プログラムに対し、単純多数決による合議を導入した。モンテカルロ法は乱数を用いた探索法のため、探索を複数回行い単純多数決をさせることで容易に合議をさせることができる。しかし、先に述べた通り乱数のみによる合議は個々のプレイヤーが似た性質を持ち、多様性が少なくなってしまう。そのため、プロによる対局の棋譜データを複数用意し、そこから個別に学習をさせることで作成した知識を用いることで多様性を増した。多様性による変化を調べるため、学習を行う棋譜群に対して同一棋譜の割合を変化させるBagging法を行うことで知識を変化させた。また、学習時に他のプレイヤーの学習結果を考慮するNCLを行い、知識に対して多様性を持つように調整を行うことで合議制囲碁プレイヤーの強化を行った。

2. 関連研究

2.1 モンテカルロ囲碁

囲碁では、駒が黒と白の2種類のみであり、打たれた手が後々良い手か悪い手か判断されるなど、途中経過の盤面に対する評価が難しいため、良い評価関数を作成することは非常に難しいといわれている。しかし、囲碁には局面において最善手以外にも良い手が多く、

単純なシミュレーションでも容易に終局状態まで到達できるといった性質もある。これは、乱数を用いてシミュレーションを行うモンテカルロ法と相性がよく、モンテカルロ法を用いたモンテカルロ碁が大きな成果をあげている³⁾。

モンテカルロ碁では、作成することが難しいとされている評価関数を、ランダムなシミュレーションによって終局まで打ち終え、最終的な勝敗を取得するプレイアウトを複数回行い作成した局面評価を用いている。単純なモンテカルロ碁では何度もプレイアウトを行い、一番勝率の高い手を選ぶ。しかし、限られたプレイアウト数では、可能な合法手全てに対して均等にプレイアウトを行うよりも、できるだけ良さそうな手に多くのプレイアウトを行う必要がある。どの手にプレイアウトを行うかという問題に対して、現在のモンテカルロ碁では主に UCT 探索が用いられている⁴⁾。

UCT 探索とは、K 腕バンディット問題で有効な手法である UCB1 アルゴリズムを木構造に拡張したものである。K 腕バンディット問題とは、腕が K 本ありそれぞれ独自の確率があるスロットマシンをプレイする問題である。プレイヤーは各腕に対する知識を持っていない状況から、プレイしながら報酬の和を最大にすることを目的としている。K 腕バンディット問題は探索と収穫のジレンマが問題であり、UCB1 アルゴリズムは腕に関する知識を増やす探索と、知識から最善の腕を選ぶ収穫を上手く行うことで、報酬の和が最大になるように腕を選ぶアルゴリズムである。UCT によって探索であるプレイアウトの数と収穫である勝率のバランスを取ることで、有効そうな手により多くのプレイアウト数を割り当てることができる。

主なモンテカルロ碁は、UCT による木探索部分とプレイアウトによるランダムシミュレート部分とで構成されている。

2.2 Elo レーティング

モンテカルロ碁に対し、知識を導入することでプレイアウトの質を上げることができることが確認されている。導入する知識には様々な種類と手法があるが、8 斤傍の石の配置をパターンとし棋譜データから出現頻度を学習し、探索局面に対して合法手の確率分布を持たせることで探索時に好手が選ばれる確率を上げる手法が、知識作成の容易さと実行速度の面で有効であり、広く使われている⁵⁾。

棋譜からの教師あり学習として Elo レーティングの一般化に基づいた手法がある⁶⁾。チーム対チームの試合結果に関する予測を行う Bradley-Terry モデルを用い、学習棋譜中の手を勝者、他の合法手が敗者の試合と考えることで、碁の教師あり学習に適用できる。特徴の集合をチームと考え、個々の特徴の Elo レーティングを計算することで、探索局面に対し

て合法手の確率分布を持たせることができる。この手法の優れた点として、チームとして考えることで特徴の個数に対してコストは指数的に増えずに、複数の特徴を組み合わせることができる。また、従来の特徴の出現率を用いた手法と違うところとして、同時に出現している打たれていない特徴を考慮している。そのため、より正確な評価を行うことができる。

Bradley-Terry モデルは、過去の試合結果から参加者一人一人の強さを推定し将来の試合結果の確率分布を与える。Bradley-Terry モデルのチーム戦における一般化の式は (1) のように表される。

$$P(1-2-3 \text{ が } 4-2 \text{ と } 1-5-6-7 \text{ に勝つ}) = \frac{\gamma_1 \gamma_2 \gamma_3}{\gamma_1 \gamma_2 \gamma_3 + \gamma_4 \gamma_2 + \gamma_1 \gamma_5 \gamma_6 \gamma_7} \quad (1)$$

この時、各個人 i の強さを正の数値 γ_i で表し、 i が強いほど γ_i は大きくなる。強さの推定は γ をパラメタのベクトル、 R を過去の結果としたとき、ベイズの公式より (2) で得られる。

$$P(\gamma | R) = \frac{P(R | \gamma)P(\gamma)}{P(R)} \quad (2)$$

パラメタ γ は $P(\gamma | R)$ が最大に成る γ^* を見つけることで推定される。仮想的な結果 R' を事前分布 $P(\gamma) = P(R' | \gamma)$ とすることで最適化すると、 γ の推定は $P(R, R' | \gamma)$ の最大化となる。 $\gamma_1, \dots, \gamma_n$ を n 人の強さパラメタとし、各個人間の独立な N 回の試合結果 $R_1, \dots, R_N$ は既知であり、試合結果の確率は 3 で書ける。

$$P(R_i) = \frac{A_{ij}\gamma_i + B_{ij}}{C_{ij}\gamma_i + D_{ij}} \quad (3)$$

ここで、 A_{ij}, B_{ij}, C_{ij} , そして D_{ij} は γ_i と独立な係数である。この式より各 $P(R_i)$ をある特定の γ_i の関数として、異なる n 通りの方法で書くことができる。 E_j は $E_j = C_{ij}\gamma_i + D_{ij}$ と定義され、最大化する目的関数は (4) となる。

$$L = \prod_{j=1}^N P(R_j) \quad (4)$$

L を最大化する反復アルゴリズムとして少数化-最大化を用いる。 γ の初期推定値 γ^0 から初め、 γ^0 で L を少数化する関数 $m(x)$ を作る。その後 $m(x)$ が最大値を取る γ^1 を求めると、少数化の性質により γ^1 は γ^0 から改善されている。 L を γ_i の関数と考え、対数を取り γ_i を含まない項を取り除く。また、 $B_{ij} = 0$ または $A_{ij} = 0$ なので、最大化すべき関数は (5) に成る。

$$f(x) = W_i \log x - \sum_{j=1}^N \log(C_{ij}\gamma_i + D_{ij}) \quad (5)$$

右辺の対数は $x = \gamma_i$ における接戦で少数化され、 x を含まない項を取り除くと、最大化する少数化関数は 6 となる。

$$m(x) = W_i \log x - \sum_{j=1}^N \frac{C_{ij}x}{E_j} \quad (6)$$

$m(x)$ の最大値は、 x について微分し、0 と置いて解いた式 (7) で得られる。

$$x = \frac{W_i}{\sum_{j=1}^N \frac{C_{ij}}{E_j}} \quad (7)$$

L を最大化するように計算を繰り返すことで、より精度が上がる。

2.3 乱数合議

乱数合議とは、複数のプレイヤーの判断を統合する合議手法の 1 つであり、単一の思考プログラムを用いて擬似的に複数のプレイヤーを作りたいときに有効な実現法である。複数のプレイヤーの生成方法は、探索局面に対する手の探索を個々のプレイヤーごとに異なる乱数系列を与えることによって変化させ、各プレイヤーのゲーム木探索を異なるものにするすることで、各プレイヤーが出す結果を異なるものにする。このようにして生成された複数のプレイヤーの最終的な判断を多数決や一番評価値が高いものを選ぶ事で決定する。

囲碁プログラムに対して乱数合議を適用し、単一の囲碁プログラムと対戦させた結果、有意に勝ち越すことが確認されている。

2.4 集団学習

集団学習とは、機械学習手法の 1 つである。複数の判別機を異なる手段により学習することで生成し、複数の判別機による出力を統合することで、最終的な判断を行う。

複数の判別機を用いる利点として、多数決で統合することで単一の判別機では誤ってしまうような場面でも、過半数以上が誤判別を行わないかぎり避けることができる。また、膨大な学習データでも各プレイヤーに割り当てることが可能である。さらに、学習を変えることで個々のプレイヤーの欠点を補い合う判別機を用いることができる。

集団学習では異なる複数の判別機をどのように生成するか、生成された複数の判別機の出力をどのように統合し最終的な出力にするかを考慮しなければならない。

2.5 Negative Correlation Learning

Negative Correlation Learning とは、最急降下法による学習において特別な目的関数を用いた学習を行うことにより、生成される複数の判別機の多様性に注目した学習を行う集団学習手法の 1 つである。この手法では、明示的に学習において判別機が多様になるように調整を行う。学習の手順として、複数の判別機の出力を (8) のように表現する。

$$\hat{f}(k) = \frac{1}{M} \sum_{i=1}^M f_i(k) \quad (8)$$

ここで、 M は判別機の数、 $\hat{f}_i(k)$ は i 番目の判別機の k 番目の学習標本に対する出力を表す。次に、最急降下法で個々の判別機を学習する際に用いる損失関数を (9) のように定義し、最小化することを考える。

$$e_i(k) = \frac{1}{2}(\hat{f}_i(k) - c(k))^2 - \alpha p_i(k) \quad (9)$$

$$p_i(k) = (\hat{f}_i(k) - \hat{f}(k))^2 \quad (10)$$

ここで、 $c(k)$ は k 番目の学習標本に対する教師例、 $p_i(k)$ は判別機間の分散が小さいことに関するペナルティ、 α はペナルティの影響度を表す係数である。 α が大きくなるほど個々の判別機の精度は落ちるが、判別機の多様性は増すという関係になっている。以上のように、教師例と近い判別機間の分散が大きくなることを目指す目的関数を定義し、学習を行うことによって多様性に注目した集団学習を実現している。

3. 本研究での手法

3.1 単純合議の囲碁プログラム

合議に用いる囲碁プログラムは、モンテカルロ法と Elo レーティングにより学習された知識を用いている。モンテカルロ法に使用される乱数系列と探索局面に対する合法手の確率分布を与える知識を変化させることで、単一のプログラムから複数のプレイヤーを作成した。合議は、それぞれのプレイヤーが探索し選択した手から多数決で手を選択する、多数決合議を行う。

3.2 Elo レーティングへの NCL の適用

Elo レーティングを利用した学習に NCL を適用することで、他のプレイヤーが持つ学習結果と離れるように学習を行い、多様性を持たせる。

Elo レーティングにおいて最大化を目指す関数 (6) に対して他のプレイヤーの学習との分散である第 2 項を付け加え、NCL を導入した (refhxin) を作成した。

$$h(x_{in}) = \frac{m(x_n)}{N_n} + \frac{\alpha}{2(n-1)} \sum_{m=1}^{n-1} (x_{in} - x_{im})^2 \quad (11)$$

このとき、 x_{in} は n 番目のプレイヤーのパターン i の評価値、 N_n は n 番目の学習棋譜群の局面数、 α はプレイヤー間の学習の距離をどれほど重視するか係数である。この式の最大値は、 x について微分し 0 と置いて解いた式 (12) となる。

$$x_{in} = \frac{\frac{1}{N_n} \sum_{j=1}^{N_n} \frac{C_{ijn}}{E_{nj}} + \alpha \bar{x}_{in} - \sqrt{(\frac{1}{N_n} \sum_{j=1}^{N_n} \frac{C_{ijn}}{E_{nj}} + \alpha \bar{x}_{in})^2 - 4\alpha \bar{w}_{in}}}{2\alpha} \quad (12)$$

ここで、 C_{ij} は j 番目の局面中でのパターン i の評価値の合計、 E_j は j 番目の局面に出現するパターン全ての評価値の合計である。さらに

$$\bar{x}_{in} = \frac{1}{n-1} \sum_{m=1}^{n-1} (x_{in} - x_{im}) \quad (13)$$

$$\bar{w}_{in} = \frac{W_{in}}{N_n} \quad (14)$$

とする。ここで、 $\bar{x}_{in}$ は n 番目までのプレイヤーのパターン i に対する学習の分散を表し、 W_{in} は n 番目の学習標本中でパターン i が打たれた数であり、 $\bar{w}_{in}$ がパターン i の使用率を表す。この時、NCL 項が大きすぎ根号中の値が負になり発散しないように、係数 α の値を十分に小さく設定する必要がある。

4. 実験

4.1 合議の作成

実験に用いる囲碁プログラムとして LibEGo を使用した⁸⁾。LibEGo は、モンテカルロ法と UTC、Elo レーティングを利用した知識が用いてあり、実行速度が早くオープンソースのため研究に向いている。

LibEGo を元に 10 プレイヤによる単純多数決合議を作成した。それぞれのプレイヤーは、モンテカルロ法に使われる乱数系列、盤面に対して合法手の確率分布を与える知識を変化させることで多様性を持たせている。

4.2 知識の作成

知識は、プロによる対局の棋譜を 1 プレイヤに対し 1,000 棋譜の計 10,000 棋譜を用いて学習した。

まず、Bagging 法として各プレイヤーが学習する 1,000 棋譜に対して同一の棋譜を持たせた学習棋譜の作成を行った。同一の棋譜は、0 棋譜、250 棋譜、500 棋譜、750 棋譜そして 1,000 棋譜の 5 種の学習棋譜を各プレイヤーごとに作成した。同一棋譜数が多くなるほど、作成される知識の各プレイヤー間での多様性が少なくなり、同一棋譜が 1,000 棋譜による合議は各プレイヤーの知識が同一となるため、乱数系列のみが違う合議となる。

学習は、学習棋譜に登場する局面に対して、合法手の 3×3 による 8 近傍の石の配置とアタリを特徴とした Elo レーティングを利用した学習を行った。

さらに、NCL を適用した Elo レーティングへ学習式を変更し、作成した学習棋譜に対して知識の作成を行った。ここで、プレイヤー間の学習の距離をどれほど重視するか係数 α は 10^{-14} とした。

4.3 実験方法

各プレイヤーはプレイアウト数を 20,000 回とした。対局は 9×9 の盤面で中国ルール、コミ 6.5 として行った。

4.4 結果

まず、知識を用いることの有用性を確かめるために知識の有無で白番 50 回、黒番 50 回の計 100 回対戦をさせた。結果を表 1 に示す。次に、合議の有用性を確かめるために単純多数決合議のプログラムと、単一プログラムで白番 50 回、黒番 50 回の計 100 回対戦をさせた。結果を表 2 に示す。

表 1 知識を適用した対戦結果		表 2 合議と単体による対戦結果	
知識有りのプログラム	知識無しに対する勝率 (%)	合議プログラム	単一プログラムに対する勝率 (%)
単一プログラム	90	知識無し	74
合議プログラム	82	知識有り	65

表 1、表 2 の対戦結果は危険度 1%における 100 戦の二項検定から有意に勝ち越していると言える。このことから、知識と合議が有効であることがわかる。

さらに、Bagging 法を用いた合議と Bagging 法で作成した学習棋譜に対して NCL を用いて学習した合議による実行結果を表 3 に示す。対局相手としてオープンソースの GnuGo を Level0 で用いた⁹⁾。対局は白番 150 回、黒番 150 回の計 300 回を行った。

表 3 Bagging と NCL の実験結果

同一棋譜数	Bagging のみの勝率 (%)	NCL を加えた勝率 (%)
0 棋譜	63.4	68.5
250 棋譜	63.0	63.9
500 棋譜	65.2	68.6
750 棋譜	63.7	66.0
1,000 棋譜	61.6	65.3

表 3 の同一棋譜数が 1,000 棋譜の Bagging は各プレイヤー間の違いが乱数のみよる合議である．この乱数のみの合議は，知識の差異による多様性を持つ他の全ての合議に比べると勝率が低い．このことから合議において集団の多様性が重要であるということが示せる．また，多様性を増すように学習した NCL を用いた合議では，個別の学習である Bagging と比べ全て勝率が上がっていることから NCL が有効であることがわかる．

5. おわりに

囲碁の合議においても，乱数のみが違う合議に対して，知識の差異による多様性を増すようにした合議の勝率が上回っていることから，多様性を増すようにプレイヤーを用意することで勝率を上げることができることが示せた．また，知識に対して多様性を増すように調整する NCL を適用することによって，さらに勝率を上げることができることが示せた．

今回は係数 α を 10^{-14} として用いたが，これはいくつかの値を試し勝率が上がった物を用いたためである．また，Elo レーティングを用いた学習では学習標本数が少なくても良い成果を出せるとのことだったが，1,000 棋譜でなくより多くの棋譜を各プレイヤーに割り当てることによってより学習の結果が多様になるのではないかと，ということ調べる必要がある．

参 考 文 献

- 1) 宮本翼:“合議制アルゴリズムを用いた囲碁の着手決定手法”，九州工業大学大学院修士論文 (2010) ．
- 2) 鈴木洋平，三輪誠，金田康正:“合議のための多様な将棋プレイヤーの集団学習”，GPW2012，pp.17-24 (2012) ．
- 3) 清 慎一，山下 宏，佐々木 宣介:“コンピュータ囲碁の入門”，共立出版，2005 ．
- 4) P. Auer, N. Cesa-Bianchi, and P. Fischer:“Finite-time analysis of the multiarmed bandit problem”，Machine Learning，47(2/3):pages 235-256，2002.
- 5) Sylvain Gelly, Yizao Wang, Remi Munos, and Olivier Teytaud:“Modification of UCT with patterns in Monte-Carlo Go”，Technical Report 6062，INRIA, France，2006.

- 6) Remi Coulom:“Computing Elo Ratings of Move Patterns in the Game of Go”，ICGA Computer Game Workshop (2007) ．
- 7) Y.Liu, X.Yao:“Ensemble learning via negative correlation”，Neural Networks，vol.12，no.10，pp.1399-1404 (1999) ．
- 8) libEGO，<http://www.mimiw.edu.pl/lew/hg/libego/>，2011 ．
- 9) GnuGo，<http://www.gnu.org/software/gnugo/gnugo.html>，2011 ．