多様性に注目した将棋プレイヤの集団学習に関する調査

鈴 木 洋 平[†1] 金 田 康 正[†1,†2]

近年，複数の将棋プレイヤによる多数決を行うことによって最終的な指し手を決定する，合議と呼ばれる手法が注目されている．合議とは，構成する複数のプレイヤを評価関数の評価値を乱数を用い変化させることによって生成する手法である．近年では，集団学習手法によって合議を形成する手法も効果があることがわかってきている．本稿では多様性に注目した集団学習による合議の方法をいくつか提案し，それらを比較した．

An investigation on ensemble learning of Shogi players focused on diversity

Yohei Suzuki[†1] and Yasumasa Kanada[†1,†2]

Consultation algorithm for shogi players has been recently focused. The algorithm creates various players from a single player by adding small random values to its evaluation function scores, and decides a move by applying majority rule to the players' decision. Recent years, it was shown that the consultation algorithm built by ensemble learning method is effective. This paper compares various methods to apply an ensemble learning method focusing on diversity of consultation to computer shogi players.

1. はじめに

近年，複数の将棋プレイヤの判断を多数決を行うことによって最終的な指し手を決定する合議と呼ばれる手法が注目されている[1)]．複数の将棋プレイヤにそれぞれ独立に判断を行わせるという比較的単純な並列化手法にも関わらず，多数決により判断ミスの少ない強い将棋プレイヤを生成できることがこの手法の長所である．小幡らは将棋プログラム bonanza に，乱数合議と呼ばれる手法を用いることによって従来の bonanza に有意に勝ち越すプレイヤを生成することに成功した[1)2)]．なお，乱数合議とは，将棋プレイヤが持つ評価関数の評価値を乱数により変化させることで複数の判断を生成し，それらを用いた多数決などで最終的な指し手を決定する手法である．これに対し複数の異なるプレイヤを生成して利用する手法は機械学習の分野において集団学習として広く研究されてきた．集団学習は，複数のプレイヤをそれぞれ異なる学習例を用いて学習することにより生成し，その判断を統合する手法である．Bagging[3)]，Boosting[9)] などの具体的な手法が存在し，パターン認識分野において多くの成果を出している[3)4)5)9)]．以上を踏まえ鈴木は集団学習の手法である Bagging，Boosting を用い，複数の将棋プレイヤを異なる学習例を用いて生成し，その結果を用いて合議を行った（アンサンブル合議）[10)]．鈴木は集団学習によって生成された合議プレイヤが，合議プレイヤの生成と同じだけの量の標本を学習した単一の将棋プレイヤに大きく勝ち越すことを確認した．さらに鈴木らは，一般に集団学習手法の精度は学習結果の散らばり具合に依存すると言われている[4)] ことをもとに，合議を構成する将棋プレイヤの評価関数のパラメータ同士の分散を大きくするよう学習の更新式を調整した集団学習によって，より強力な合議プレイヤを生成することで集団学習による合議プレイヤのさらなる改良の可能性を示した[11)]．しかし，鈴木らの手法では評価関数のパラメータの散らばりを大きくするように学習を行うが，評価関数の多様さが直接手の多様さにつながる保証はなく，実際に何が合議プレイヤの違いを生み出しているのかを解明することはできなかった[11)]．

上記を踏まえ，本稿では，将棋プログラムの多様性に注目した集団学習手法によって生成される合議プレイヤについて，様々な角度から詳細な解析を行っていく．具体的には，Bagging に加え合議を構成する各プレイヤの分散を考慮した様々な目的関数を用いて多様性が増すようにしながら複数のプレイヤを作成し，合議させることで合議の挙動を分析する．合議プレイヤの多様性について，様々な視点から考えることができる．そこで，多様性を評価するいくつかの指標を導入することにした．本稿では三つの多様性の指標につい

†1 東京大学工学系研究科
Graduate School of Engineering, The University of Tokyo
†2 東京大学情報基盤センター
Information Technology Center, The University of Tokyo

て定義し，議論する．．一つ目は，指し手の候補として個々の将棋プレイヤが挙げる手の種類である（以降では候補手数と呼ぶ）．この数値が大きいほど，多様な手が指し手の候補として挙げられていることになる．二つ目は，熟練者の棋譜を用いて検証を行った時に，合議プレイヤを構成する個々の将棋プログラムが挙げたいずれかの手が熟練者の選択した手と一致する確率である（以降ではカバレッジと呼ぶ）．カバレッジが大きいほど，合議プレイヤが正解を選択できるポテンシャルが高いと考えられ，一つ一つの将棋プレイヤの重要度・自信度を与える学習手法と併用すれば強力な合議プレイヤを生成できる可能性がある．三つ目は，鈴木らが行った実験と同じように[11]，評価関数同士のパラメータの分散である．評価関数同士の分散が大きければ，局面に対する評価関数の値も異なってくるので，多様な判断が行える可能性がある．

本論文では以降，2 章において関連研究について述べたのち，3 章で提案手法について述べ，4 章では提案手法を用いての評価について述べ，5 章でその結果を考察する．最後に 6 章でまとめと今後の課題を述べる．

2. 関 連 研 究

2.1 乱 数 合 議

乱数合議とは，複数の将棋プレイヤの判断を統合する合議手法の一つであり，単一の思考プログラムを用いて複数のプレイヤを作りたいときに有効な合議の実現法である[1]．乱数合議の仕組みは**図 1** のように表現できる．複数のプレイヤを生成する方法は，将棋プログラムにおける評価関数の探索局面に対する評価値を，個々のプレイヤごとに異なる乱数系列を与えることによって変化させ，各プレイヤのゲーム木探索の結果を異なるものにするというものである．評価関数に与える乱数は，正規分布 $N(0, D^2)$ にしたがって生成し，局面のハッシュキーに割り当てる．これにより，個々のプレイヤの評価関数は同じ局面に対しては常に同じ評価値を算出する．このようにして生成された複数のプレイヤによる多数決 (あるいは最も高い評価値を算出したプレイヤの指し手[2]) で指し手を決定する．

小幡らは Bonanza[7] に対しこの乱数合議法を適用し，一手あたりのノード数を固定した上で単一の通常の Bonanza と対戦させ，合議のプレイヤ数や正規乱数の分散値によっては有意に通常の Bonanza に対し勝ち越すことを確認した[1]．

2.2 集 団 学 習

集団学習とは，機械学習手法の一つであり，複数の判別を行うユニット (以降は判別器と呼ぶ) を生成し，それらの判断を統合した上で最終的な判断を生成する手法のことである．手順としては，まず学習の段階において，複数の判別器を異なる手段により学習することで生成する．次に，実際の問題に対する判断を生成する際，複数の判別器による出力を適当なルールによって統合し，最終的な判断を行う．

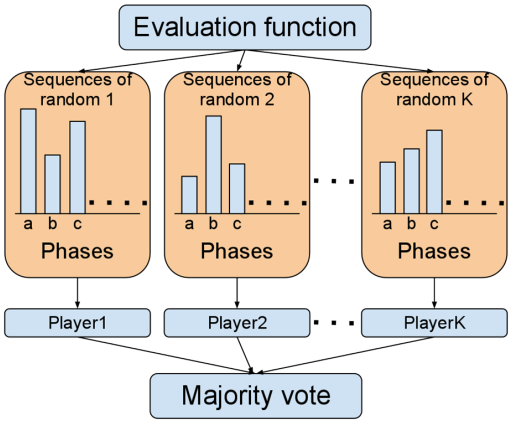

図 1 乱数合議

複数の判別器を利用する利点はいくつか存在するがそのうちの三点を述べる．一つは， 異なる複数の意見を統合することで単一の判別器では誤ってしまうような場面を避けられる可能性があることである．例えば，5 つのそれぞれが異なる判別器を用意し，それらの出力を多数決することによって最終的な出力を決定するとする．すると，3 つ以上の判別器が誤判別を行わない限りは最終的な出力が誤りになることはないので出力が安定する．二点目の利点は，複雑な標本空間を分割し，各プレイヤに割り当てることが可能な点である．三点目の利点は，個々のプレイヤの欠点を補い合うことができる点である．

集団学習の手法を設計する際考慮すべき点は主に二つある．どのように異なる複数の判別器を生成するか，どのように生成された各判別器の出力を統合し最終的な出力にするかの二点である．これらの観点から考え出された，様々な集団学習手法が現在提案されているが，Bagging もそのような手法の一つである．

2.2.1 Bagging

Bagging とは，Bootstrap Aggregating の略であり，ブートストラップ法により生成された分類器を結合するというアルゴリズムである．その手順は**図 2** のようになる．ブートストラップ法とは，与えられた学習用標本集合から複数の標本集合の複製を生成する手法である．具体的には，与えられた標本集合から，重複を許して標本のサンプリングを行い，新たな標本集合を構成する．つまり，元の標本集合の標本が何度も新しい標本集合の要素に選ばれてしまう可能性がある一方で，逆に一度も選ばれない標本があるかもしれない．この方法によって少し異なる複数の標本が得られる (図 2 の復元抽出された標本)．それらを用いてそれ

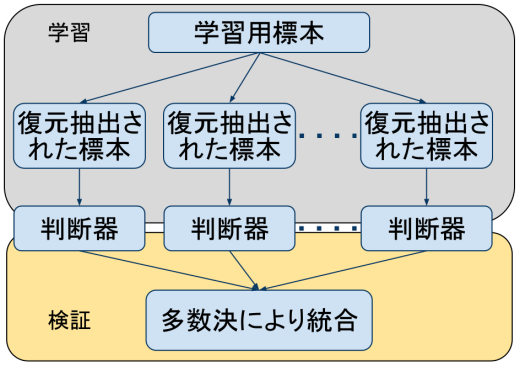

図 2　Bagging の仕組み

ぞれ判別器を学習を行い，その結果生成された複数の分類器の出力を多数決により統合する．単純に感じられるが，この方法によって多くの場合，単一の判別器よりも高い精度が得られる．理由としては，学習結果の分散が減ることが挙げられる．ここで分散とは，判別器の学習に用いる標本の違いが生成される判別器にどの程度影響してしまうかを表す．一般に，Bagging は不安定な学習プログラムに有効であるとされているのもこの理由からである．また，学習用データの少ない場合も，ブートストラップ法によりデータサイズを維持したまま複数の学習用標本集合を生成できるので，Bagging が有用である．

2.2.2　Negative Correlation Learning

Negative Correlation Learning(NCL)[5] 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で，生成される複数の判別器の多様性に注目した学習を行うという，集団学習手法の一つである．Bagging では，異なる判別器が生成されるかどうかは復元抽出による学習標本の違いに依存したが，NCL では明示的に学習において判別器が多様になるよう調整する．NCL の手順は以下のようになる．はじめに，複数の判別器 (以後アンサンブルと呼ぶ) の出力 $\hat{f}$ を式 (1) のように表現する．

$$\hat{f}(k) = \frac{1}{\mathrm{M}} \sum_{i=1}^{\mathrm{M}} \hat{f}_i(k) \tag{1}$$

ここで，M はアンサンブルを構成する判別器の数，$\hat{f}_i(k)$ は i 番目の判別器の k 番目の訓練用標本に対する出力を表す．次に最急降下法で個々の判別器を学習する際に用いる損失関数を式 (2)，式 (3) のように定義し，式 (2) を最少化することを考える．

$$e_i(k) = \frac{1}{2}(\hat{f}(k) - c(k))^2 + \lambda p_i(k) \tag{2}$$

$$p_i(k) = -(\hat{f}_i(k) - \hat{f}(k))^2 \tag{3}$$

ここで，$c(k)$ は k 番目の訓練標本に対する教師例，$p_i(k)$ はアンサンブルの分散が小さいことに関するペナルティ，λ はペナルティの影響度を意味する係数を表している．λ が大きくなるほど個々の判別器の精度は落ちるが，判別器の多様性は増すというトレードオフが存在する．

以上のように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．NCL をニューラルネットワークに適用した場合，扱う問題や条件によっては Bagging などの代表的な集団学習手法に勝る結果が得られている[6]．

3.　提 案 手 法

3.1　将棋プログラムに対する Bagging の適用

集団学習はアルゴリズムの設計に際して，学習用標本数，プレイヤ数，個々のプレイヤの複雑さ，モデルの多様性，個々の学習用標本の重要度，判断を統合する方法など，様々な観点が存在し，それらのパラメタの調整の手法が議論されている．パラメタの多さはアンサンブルの設計が複雑で曖昧になる危険をはらんでいる．しかし，調整する部分が多いことから，設計の目的に沿ったアンサンブルの調整を行うことが比較的容易にできる．

近年，将棋プログラムの評価関数を生成する際，ボナンザメソッド等[7] の熟練者の棋譜を教師例とする学習手法を用いることが一般的であり，集団学習を将棋プログラムに適用することは可能である．さらに，適切な設計方法を用いて集団学習を調整することができれば，より強力な合議プレイヤが生成できる可能性が高い．

本提案手法では，学習用の棋譜を用いて集団学習の代表的手法である Bagging による学習を行い複数の評価関数を生成し，その結果得られたプレイヤを用いて合議をする手法を提案する．さらに集団学習による合議の調整の一例として，プレイヤの多様性に注目した学習について提案する．具体的には，用意された棋譜から，個々のプレイヤに対して重複を許して決まった数の棋譜を抽出する．抽出された棋譜を用いてプレイヤの評価関数を棋譜をもとに学習する．なお，学習手法としては激指の手法[12] を用いた．激指では，ボナンザメソッド[7] と Averaged Perceptron[13] を基本にした学習手法が用いられている．ボナンザメソッドは，棋譜の手筋を再現するように評価関数を最適化する手法で，探索を用いて局面に対する評価値を算出する評価関数のパラメタを，各局面において棋譜の手が他の手に比べて相対的に高く評価されるように調整する．Averaged Perceptron はオンライン学習の一種で，学習途中の各ステップにおける評価関数重みベクトルの総和を保持し，これを平均したものを最終的な更新値にする手法である．Averaged Perceptron では重みベクトルの平滑化を行うことで，ノイズに強い学習を実現している．各ステップでの具体的な重みベクトルの

初期パラメタ:
評価関数の重みベクトルの初期値 $\boldsymbol{w}_0$
学習に用いる棋譜の絶対数 N
一つのプレイヤの学習に用いる棋譜数 M
プレイヤ数 P
各プレイヤに共通な学習に用いる棋譜数 K(< M)

学習:
(1) 熟練者の棋譜 N 棋譜から M 棋譜を復元抽出する.
(2) 重みの初期値 $\boldsymbol{w}_0$ をセット
(3) M 棋譜を使用
(4) 各プレイヤの評価関数を棋譜を用いて学習する.
(5) (1)〜(4) を P 回繰り返す

テスト:
- 学習の結果得られた P 個のプレイヤがそれぞれ探索によって指し手を決定. それらをもとに多数決によって最終的な指し手を決定.

図 3 将棋への Bagging の適用の手順

$\boldsymbol{w}$ の更新は式 (4)

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \frac{1}{|\mathrm{M}|} \sum_{j \in \mathrm{M}} (\boldsymbol{\phi}(t_1) - \boldsymbol{\phi}(t_j)) \qquad (4)$$

ここで，t_j は j 番目の合法手の後の探索における最善応手手順後の局面（t_1 は棋譜の手をあ表している），$\phi(t_j)$ は局面 t_j の特徴ベクトル，M は式 (5) を満たす合法手 j の集合である.

$$\boldsymbol{w}^T \boldsymbol{\phi}(t_1) - \boldsymbol{w}^T \boldsymbol{\phi}(t_j)) < margin \qquad (5)$$

式 (5) が満たされるときは棋譜の手が j と比べて悪く評価されてしまったことを表す. $margin$ は閾値で，激指では終盤になるほど大きな値が設定される. 以上のような学習を必要なプレイヤの数だけ繰り返すことによって，複数の異なるプレイヤを生成する. 実際に対戦を行う際は，生成されたプレイヤを用いて多数決を行い，最も支持された指し手を最終的な指し手とする.

以上を踏まえると，手順は**図 3** のようになる.

3.2 将棋プログラムに対する NCL の考え方の適用

将棋プログラムに Bagging を適用した上で，さらに生成される複数のプレイヤの多様性が増すように学習を調整する.

本稿では，学習の更新式にプレイヤの様々な多様性に関する項を加えることによって，合議プレイヤがどのような挙動をするのかを解析する. 通常の NCL では，式 (2)，式 (3) のように，アンサンブルの各プレイヤの多様性が増すように，各判別器の訓練標本に対する出力の分散をペナルティ項として用いていた. しかし，将棋プログラムは，各プレイヤの正確な出力（つまり評価値や，指し手) を導出するためにゲーム木を用いた探索を行なわなければならず，これをアンサンブルを構成している各プレイヤに対し行えば膨大な計算量が必要になる. 浅い探索や，局面の静的評価を用いて式 (3) のようなペナルティ項を導入することはできるが，今回は簡単のために評価値以外の新たな多様性の指標を複数用意し，それらを用い評価関数の学習を行う. 以上のように，本提案手法は正確には NCL[5] と異なる.

学習の手順としては**図 4** のようになる. ブートストラップ法による復元抽出により評価関数を学習するための標本を用意し，合議の個々のプレイヤの多様性が増すように更新式を設定し評価関数を学習する. しかし，合議の多様性にも様々な指標があるので， 何がもっとも合議の多様性を表現するのか，何がもっとも合議の強さと関係しているのかの判断は難しい.

そこで，多様性の様々な指標に注目した集団学習の手法をいくつか考え，比較検討することにした. 合議の多様性を示す指標は数多く考えられる. 例えば，完全に収束した評価関数と異なる指し手を指すかどうかや，多数決で決まった指し手が実は最も評価値が高い手ではなかった確率（つまり多数決による合議と楽観合議で指し手が違ってしまう確率），多数決で最も支持された手の得票数（着手が圧倒的な支持で選ばれた手か，競った手かがわかる. 競ったほうが多様性に資すると思われる）等の多様性に関する指標を考えることもできる. さらに，探索の過程の違いを多様性の指標にすることもできると思われる.

本稿では，評価関数同士の距離，候補手数，カバレッジの三つの多様性の指標について考えることにした. 評価関数同士の距離を考える方法は，以前に集団学習の合議への適用を議論した際[11] にも用いた考え方で，利点としては評価関数同士の距離を測るだけで学習の更新が行えるので，他の多様性に注目した集団学習手法と比べると学習コストが少ないことがあげられる. 将棋プログラムの判断の基準は評価関数による評価値であり，その評価値は評価関数の重みベクトルによって値が変動するので，判断は多様になることが期待される. 候補手数とは，各局面に対して合議の集合の中で何通りの候補手が検討されるかを表す. この数が多ければ多いほど，様々な意見が出てきているはずなので多様な合議といえる. 候補手が多いことが必ずしもいいとは限らない. 例えば，8 プレイヤの合議で 8 通りの候補手が出るような状況は，まともな多数決ができないので望ましくない. しかし，実際にそのようなことはほとんどおこらない. 後述するが，候補手の一局面あたりの平均は NCL を用いない通常の Bagging を用いて生成した合議プレイヤ (8 プレイヤ) で 1〜2 程度である. 乱数合議においても 3 を超えない程度の候補手数しかでない. このことから，合議においては普段一つの候補手が圧倒的に支持されて選択される状況がほとんどであると思われる. もちろんミスを減らすという意味が十分にあるが，これでは合議のポテ

ンシャルはほとんどない．よって，候補手を増やすよう学習することに価値がある可能性がある．カバレッジとは，棋譜の手が合議の中で候補手として挙がった確率を表す．もし多数決の結果選ばれた手が棋譜の手ではないとしても，棋譜の手を挙げたプレイヤがひとつでもあるのであればカバレッジに計上される．このカバレッジの値が大きいほど，合議プレイヤが正解する（棋譜と同じ手を選択する）ポテンシャルを持っていることになる．本稿では多数決によって合議の最終的な指し手を決定しているが，今後もっと正確に正解を導く指し手やプレイヤを選択する枠組みを考える時に，カバレッジが高い数値を持っていることは望ましい．よって，カバレッジを大きくするような学習を行うことにも意義があるだろう．候補手数やカバレッジを考えた学習を行う場合，各局面に対して合議を構成するプレイヤの指し手を導出する必要があるので，評価関数の重みベクトル同士の距離を測る場合よりも学習コストがかかる．本提案手法では，逐次で評価関数を生成していくので学習が後半になるほど評価関数の重みベクトルの更新に時間がかかってしまうことになる．本稿の実験ではプレイヤ数を 8 としたが，これを 16，32 と増やしていこうと考えたときにさらなる工夫が必要になるだろう．

以上のような三つの指標を用いて今回四つの学習手法を定義した．式 (6)〜式 (9) がそれぞれの学習手法に用いる更新式である．

$$w^t_{i,new} = w^t_{i,old} + \frac{\alpha}{|\mathrm{M}|}\sum_{j\in \mathrm{M}}(\boldsymbol{\phi}(t_1)-\boldsymbol{\phi}(t_j)) + \lambda(w^t_{i,old}-\bar{w}^t) \tag{6}$$

$$w^t_{i,new} = w^t_{i,old} + \frac{1}{|\mathrm{M}|}(\alpha_0 + \frac{\beta}{1+candidate})\sum_{j\in \mathrm{M}}(\boldsymbol{\phi}(t_1)-\boldsymbol{\phi}(t_j)) \tag{7}$$

$$w^t_{i,new} = w^t_{i,old} + \frac{1}{|\mathrm{M}|}\sum_{j\in \mathrm{M}}(\boldsymbol{\phi}(t_1)-\boldsymbol{\phi}(t_j)) + \frac{\lambda w_{i,old}}{1+candidate} \tag{8}$$

$$w^t_{i,new} = w^t_{i,old} + \frac{1}{|\mathrm{M}|}\sum_{j\in \mathrm{M}}(\boldsymbol{\phi}(t_1)-\boldsymbol{\phi}(t_j)) + \frac{\lambda w_{i,old}}{1+coverage} \tag{9}$$

なお，$w^t_{i,new}, w^t_{i,old}$ は i 番目に生成する評価関数の重みベクトル $\boldsymbol{w}$ の t 行目の項の更新後と更新前の値をそれぞれ表す．$candidate$ は局面に対して i 番目以前（つまり，$i=1,...,i-1$．$i=1$ のときは通常の式 (4) のような Averaged Perceptron による更新を行う）の評価関数が挙げた候補手の個数， $coverage$ 関数は局面に対して以前の評価関数が挙げた候補手が学習に用いた棋譜の手と一致している個数を表す． それ以外の部分は，式 (4) と同じ，Averaged Perceptron の更新式である．式 (6) は評価関数の重みベクトルに注目し，重みベクトル同士の距離が離れるように学習するよう定義された更新式である．式 (7)，式 (8) は候補手数に注目し，候補手数が大きくなるように定義された更新式である．式 (9) は，カバレッジを最大化するように調整された更新式である．比較のため，式 (6) を Method1，式 (7) の手法を Method2，式 (8) の手法を Method3，式 (9) の手法を Method4 と呼ぶことにする．

この式を評価関数の更新の際に用い，各プレイヤの評価関数を，以前に生成された評価関数を用いて更新していく．その結果得られた複数のプレイヤを用いて，実際の対戦において多数決合議を行い，指し手を決定する．

初期パラメタ:
評価関数の重みベクトルの初期値 $\boldsymbol{w}_0$
学習に用いる棋譜の絶対数 N
一つのプレイヤの学習に用いる棋譜数 M
プレイヤ数 P
プレイヤの分散の程度を決定する係数 λ

学習:
(1) 熟練者の棋譜 N 棋譜から M 棋譜を復元抽出する．
(2) 重みの初期値 $\boldsymbol{w}_0$ をセット
(3) M 棋譜を使用
(4) 各プレイヤの評価関数を棋譜を用いて学習する．
(5) (1)〜(4) を P 回繰り返す
プレイヤの多様性に関する項を加えた更新式を用いて学習をする．多様性の度合いを λ によって調整する．

テスト:
- 学習の結果得られた P 個のプレイヤがそれぞれ探索によって指し手を決定．それらをもとに多数決によって最終的な指し手を決定．

図 4 将棋への MNCL の適用の手順

4. 評　　価

4.1 評 価 設 定

4.1.1 学習の設定

本手法の検討ために，将棋プログラム「激指[8]」を用いて実験を行った．激指は，第 20 回世界コンピュータ選手権において優勝を収めている，トップレベルの将棋プログラムの一つである．本研究でプレイヤを学習する際は，激指において用いられている評価関数の特徴を用いる．学習前の評価関数の重みの初期値は激指に元々用いられていた重みパラメタとし，これに対し学習を行うことによって性能評価を行った．

本提案手法では，棋譜学習の際のゲーム木の探索深

さを 6 として実験を行っている．また，学習用に熟練者の棋譜 20,000 棋譜を用意し，この 20,000 棋譜から復元抽出することで各プレイヤの学習用棋譜を生成する．

4.1.2 検証の設定

一致率やカバレッジ等の数値を出すために，学習用に用意した 20,000 棋譜とは別の 500 棋譜を用意し検証に用いた．検証をする際の探索深さは学習時と同じ 6 とする．

4.1.3 対戦実験の設定

対戦の初期局面は用意したテスト用棋譜の最初の 30 手が進行した後の局面とし，ひとつの初期局面に対し先手後手を入れ替えて対戦することを繰り返し勝率を求めた．対局は 1,000 局ずつ行った．一手あたりの一プレイヤあたりの探索深さを 6 に固定した．

4.1.4 比較用の単一のプレイヤ

集団学習と通常の学習の精度を比較するため，激指の元々の評価関数の重みを単一のプレイヤとして用意した．以降，この単一のプレイヤを単に比較手法と呼ぶ．

4.1.5 比較用の乱数合議プレイヤ

集団学習による合議と乱数合議を比較するための乱数合議を用意した．激指の元々の評価関数の重みを使い，局面の評価値に与える正規乱数の分散値を 1 プレイヤの激指に対する勝率が最も高い部分に調節した．プレイヤ数は 8 とする．

4.2 将棋への Bagging の適用

激指に Bagging を適用して学習を行い，得られた合議プレイヤの精度を確かめた．手順は図 3 のようになる．本実験では比較手法と同じように N=20,000，M=5,000 として実験を行う．さらにプレイヤ数 P=8 とした．プレイヤ数も集団学習において重要な要素といえるが，本提案手法は学習用標本と学習の目的関数を調整することによって合議を強化することを目的としているので，今回は固定した．

4.3 アンサンブル合議と乱数合議の比較

予備実験として Bagging を用いた合議や乱数合議の結果，前述の多様性の指標や学習の精度がどの程度になるのかを熟練者の棋譜 500 棋譜を用いて評価したものが**図 1** である．平均一致率とは，合議を構成する個々の将棋プレイヤの棋譜の手に対する一致率を平均したものである．合議を構成する個々のプレイヤがどの程度の精度を持っているのかを表現している．候補手数とは，一局面ごとに挙げられた候補手の平均である．図 1 から，合議を構成する一つ一つの将棋プレイヤの一致率はアンサンブル合議のほうが高く，その結果合議プレイヤ全体でも乱数合議よりも若干高い一致率が得られた．しかしその反面，乱数合議の局面あたりの候補手数は 2.89 と非常に高く，合議をする前の段階ではかなり多様な手が挙げられていることがわかる．カバレッジもアンサンブル合議より 10%近く上回っていて，正しく候補手の選別ができればさらに一致率が上がる余地があることを示している．強さではアンサンブル合議のほうが上回っているので（危険度 0.05 の二項検定で有意に乱数合議に勝ち越すことが確認されている[11]），多様性ばかりがあっても強い合議プレイヤを作ることはできないということがいえる．平均一致率をある程度高く保ちながら，候補手数やカバレッジ，あるいは評価関数のパラメータの分散を大きくしていくことを模索したい．

4.4 将棋プログラムへの MNCL の適用

将棋プログラムの集団学習を，プレイヤの多様性に注目するよう調整する．手順は図 4 のようになる．基本的には図 3 と同じだが，学習の (4) の段階で用いる目的関数を通常の式 (4) ではなく，式 (6)〜式 (9) に従うものとしている．すなわち，各プレイヤの評価関数の多様性が増加するように学習する．式 (6)〜式 (9) にある $\alpha = 1, \alpha_0 = -1, \beta = 4$ とし，λ の値は

$\lambda = 10^{-5}, 5 \times 10^{-6}, 10^{-6}$ 結果は**図 2** のようになる．

なお，太字は危険度 0.05 の二項検定で有意に勝ち越していることを意味する．図 2 を見ると，Method1〜4 は，平均一致率を高く保ち，一致率もある程度高い水準を維持していることがわかる．ただし，候補手数やカバレッジなどの多様性を示す数値はほとんどアンサンブル合議のものを上回っているものの，それは軽微で，乱数合議の数値に大きく下回る．その結果，単一の将棋プレイヤに対する勝率はアンサンブル合議のものと大きく異なることはなくなってしまっている．原因としては，学習の初期値にすでに収束していると思われる将棋プログラムに元々ある評価関数を用いたことや，学習棋譜数が足りないこと，パラメータの設定が適切でないことが考えられる．それでも評価関数同士の距離に注目した Method1 は単純な Bagging 手法に比べ，合議を構成する評価関数の重みベクトルの平均ベクトルからの分散は，Method1 のほうが Bagging のものより 1 割大きいことが確認されている．また，Method2 は他の集団学習手法と比べ若干候補手数が多く，勝率も比較的高いものになっている．

なお，過去に行われた同じような実験[11] の Bagging や乱数合議の勝率や一致率等の数値が今回の実験のものと異なるが，これは評価関数の初期値（駒割のみと激指元々の評価関数），対局数 (200 局と 1,000 局)，対局条件（持ち時間固定と深さ固定，対戦相手の違い），学習に用いた棋譜の数が異なるためである．このうち，評価関数の初期値，対局条件と学習に用いた棋譜の数の違いには大きな意味があると考えられる．費やされるリソースの量が大きく異なるからである．初期

表 1 合議の正確さと多様性

	一致率	平均一致率	候補手数	カバレッジ
Bagging	40.4%	39.9%	1.56	50.3%
乱数合議	39.0%	34.9%	2.89	62.1%

表 2　提案手法の精度と多様性の指標

	単純な Bagging 手法	乱数合議	Method1	Method2	Method3	Method4
一致率	40.39%	39.00%	40.35%	40.02%	40.33%	40.57%
平均一致率	39.87%	34.93%	39.81%	39.29%	39.79%	39.96%
候補手数	1.56	2.89	1.59	1.75	1.57	1.60
カバレッジ	50.31%	62.05%	51.27%	52.63%	50.39%	51.59%
比較手法に対する勝率	**55.90%**	52.11%	51.11%	**57.05%**	**54.59%**	**54.07%**

表 3　合議の挙動に関する統計

	単純な Bagging 手法	乱数合議	Method1	Method2	Method3	Method4
対戦実験をしているときの各手法の候補手数	1.77	2.67	1.81	2.07	1.81	1.94
検証で不一致時のカバレッジ	18%	38%	18%	21%	17%	19%
占有率	84%	83%	85%	78%	83%	80%

値が激指の元々の評価関数であるなら，事前に多くの学習が行われていることになるし，対局条件が違えば探索されるゲーム木のリーフ数は大きく異なる．こういった対局条件や棋譜数の違いによる手法の効果の違いについても論じていくべきだが，3 章で述べたとおり集団学習の合議を学習によって生成するためにはかなりの時間がかかり，多くのパラメータを考えるのは難しい．

5. 考　　察

Bagging 手法や乱数合議，Method1〜4 の合議に関する様々なデータを取って違いを考察する．今回の提案手法では利用しなかったものの，合議の挙動に関する指標は様々なものが考えられる．それらの中で違いがみえる数値を選んで並べたものが**図 3** になる．

対戦実験をしているときの各手法の候補手数は，検証用棋譜で棋譜の手との一致率を見るときではなく，実際に激指と対戦実験を行っているときの候補手数である．図 2 の候補手数と比べると，乱数合議の候補手数がほぼそのままであるのに対し，他の集団学習合議の候補手数は増えている．対戦実験で生じるような局面は，コンピュータ同士の対戦なので熟練した人間の記録である学習用棋譜にも検証用棋譜にもないような場面が多く登場することが原因として考えられる．

検証で不一致時のカバレッジは，検証用棋譜で一致率を計算する際，棋譜の手と合議が導き出した手が異なっているとき，合議を構成するプレイヤの中には棋譜の手を挙げたプレイヤがいた確率を表す．つまり不正解時でもうまくすればこのカバレッジの分だけ正解できたことになる（実際にはかなり難しい）．乱数合議はこのカバレッジが約 4 割にも上り，他の手法でも 2 割近くある．将棋プログラムが大体 6 割の問題を間違えることを考えると，この数値は無視できないものである．

占有率とは，多数決合議で採決された手を，全プレイヤのうちいくつのプレイヤが挙げたかの割合を局面について平均したものである．この値が 100% に近づくほど圧倒的な得票数で候補手が選択されたことになる．これを比較すると，集団学習手法と乱数合議で大きくその数値が変わらない．この事実と乱数合議の候補手数が他の集団学習手法と比べて多い事実とを合わせて考えると，乱数合議は様々な候補手が出てくるものの，それらのほとんどは圧倒的な少数派で選ばれることは少ないということが予測できる．Method2 は他の手法に比べ占有率が低いが，Method1 等と似たようなアプローチで評価関数の生成を行っているにも関わらず，それにしては候補手数が多いことが影響していると思われる．

以上のように合議による指し手の決定はまだ多くのポテンシャルを持っており，改善の余地があることがいえる．

6. おわりに

本稿では，様々な観点からプレイヤの多様性に着目した更新式を用いつつ，将棋プレイヤの学習に集団学習手法を適用し，合議プレイヤを生成する手法の比較をおこなった．実験では，Negative Correlation Learning の考え方を Bagging による将棋の集団手法に適用し，三つの多様性の指標から学習の更新式を定義した．実験の結果図 2 のようになり，多くの手法は単一プレイヤの激指に有意に勝ち越した．しかし，実際に合議の多様性を大きく向上させるところまでは至らなかった．だが，様々な合議の挙動に関するデータを通して合議の可能性が明らかになった．

今後の課題は二点挙げられる．一つは，パラメタチューニングの徹底である．本研究では集団学習に用いるパラメタを細かく設定することはなかったが，学習率 α の値と学習経過による変動を調整する，多様性にかかわる係数 λ を大きな値に設定しながら学習が破たんしないような制約条件を導入するなど，細かな調整が必要である．

二つ目の課題は，より洗練された更新式の発見，多様性の指標の発見である．今回用いた更新式は，Averaged Perceptron の更新式を多少改変したもので，そこに理論的保障はない．NCL 自体がヒューリスティックスなので仕方がない部分はあるものの，最適化手法

により厳密にのっとった更新式を発見したい．多様性の指標に関しても，今回の候補手数やカバレッジなどの指標を用いると学習に非常に時間がかかることがわかった．時間が短縮でき，さらに有効な多様性の指標を見出すことでさらに集団学習を用いた合議の研究が効率的に行えるだろう．

参 考 文 献

1) 小幡拓弥， 杉山卓弥， 保木邦仁， 伊藤毅志. 将棋における合議アルゴリズム:既存プログラムを組み合わせて強いプレイヤを作れるか?. The 14th Game Programming Workshop， pp. 51-58， 2009.
2) 杉山卓弥， 小幡拓弥， 保木邦仁， 伊藤毅志. 将棋における合議アルゴリズム - 評価値を用いる効果について -. The 14th Game Programming Workshop， pp. 59-65， 2009.
3) Breiman， L. Bagging Predictors. Machine Learning 24， pp. 123-140， 1996.
4) Geoffrey R. Factors Affecting Boosting Ensemble Performance on DNA Microarray data. Neural Networks (IJCNN)， The 2010 International Joint Conference on， On page(s): 1 - 7， Volume: Issue: ， 18-23， 2010.
5) Y. Liu， X. Yao. Ensemble learning via negative correlation. Neural Networks，vol. 12，no. 10， pp. 1399-1404， 1999.
6) Lofstrom. T， Johansson. U， Bostrom. H. Comparing methods for generating diverse ensembles of artificial neural networks. Neural Networks (IJCNN)， The 2010 International Joint Conference on， On page(s): 1 - 6， Volume: Issue: ， 18-23 ，2010
7) 保木邦仁. 局面の学習を目指した探索結果の最適制御. The 11th Game Programming Workshop，pp. 78-83， 2006.
8) 将棋プログラム「激指」のページ. http://www. logos. ic. i. u-tokyo. ac. jp/gekisashi/.
9) Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on- line learning and an application to boosting. In Computational Learning Theory: Eurocolt ’95, pages 23—37. Springer-Verlag, 1995.
10) 鈴木洋平. 将棋プログラムに対する集団学習の適用. 東京大学工学部卒業論文，2010.
11) 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の集団学習. 第 16 回ゲームプログラミングワークショップ，pp. 17-24，2011.
12) 鶴岡慶雅. 選手権優勝記 -激指の技術的改良の解説-. 情報処理，vol. 51，no. 8，pp. 1001-1007，2010.
13) Michael Collins. Discriminative training methods for hidden markov models: thiory and experiments with perceptron algorithms. EMNLP ’02， pp. 1-8. Association for Computational Linguistics， 2002.