

将棋の評価関数の学習に有用な局面の自動選択

川上 裕生^{1,a)} 浦 晃² 三輪 誠³ 鶴岡 慶雅² 近山 隆²

概要：将棋プログラムの評価関数は大量の棋譜を利用した機械学習によって調整する。これにはプロ棋士の棋譜が用いられているが棋譜の数には限りがあり、新たに指し手の教師情報のついた局面を作成するには大きなコストが必要となる。本稿では、教師情報を付けるコストを削減するために、能動学習を用いて学習に有効に働く局面を選択する手法を提案する。既存の棋譜を用いて提案手法の評価を行ったところ、将棋の評価関数の学習に有効な局面が存在し、その選択が可能であることを示した。

Automatic Selection of Useful Positions for Learning Shogi Evaluation Functions

YUSEI KAWAKAMI^{1,a)} AKIRA URA² MAKOTO MIWA³ YOSHIMASA TSURUOKA² TAKASHI CHIKAYAMA²

Abstract: Today, the parameters of evaluation function of a shogi program are usually adjusted by machine learning methods using many game records. This approach requires game records of professional Shogi players, but the number of such game records is limited and creating training data is costly. In this paper, to reduce the cost of creating training data, we propose an active learning-based method to select positions that are particularly useful for learning. We evaluated the proposed method by using existing game records. Experimental results show that there indeed exist such positions and that it is possible to select them automatically.

1. はじめに

人工知能の研究の一分野として、ゲームは古くから研究されてきた。中でも将棋やチェス、囲碁といった二人零和有限確定完全情報ゲームは、コンピュータプレイヤーとの相性のよさからさかんに研究が行われている。

本研究ではその中で将棋を対象とする。現在コンピュータ将棋プログラムの多くは木探索と評価関数を用いたアルゴリズムを利用しており、コンピュータ将棋の棋力を向上させる手段の一つとして評価関数の精度を向上させることが考えられる。評価関数とは、局面の良し悪しを示す評価値を局面の特徴から計算する関数である。評価関数を作成

するときは、駒の損得や駒の働きなどの特徴を局面から抽出し、それらに適切な重み付けを行う必要がある。かつて評価関数はシステム開発者の経験や勘に基づいて作られていたため、システム開発者の棋力により精度が左右されたが、現在ではプロ棋士の棋譜を利用して重み付けを機械学習により自動的に調整するという手法が広く用いられている。しかし教師となる指し手のついた局面（教師データ）の生成にはプロ棋士のような強力なプレイヤーが指し手を考える必要があるため、教師データを増やすには大きなコストが必要となる。コンピュータを用いて教師データを自己生成することによってコストを削減するという手法もある [1] が、1局面ごとに長い時間をかけて探索する必要があり、大量に教師データを作成することは難しい。

本論文では学習に有効に働く局面を能動学習を利用することによりプログラムが自動的に選択する手法を提案する。能動学習により、少量の教師データの追加で学習の効果が高くなることが期待され、少ないコストで精度の高い評価関数を作成することが可能になると考えられる。

¹ 東京大学工学部電子情報工学科

Department of Information and Communication Engineering, Faculty of Engineering, The University of Tokyo

² 東京大学大学院工学系研究科

Graduate School of Engineering, The University of Tokyo

³ マンチェスター大学コンピュータ科学科

School of Computer Science, The University of Manchester

a) kawayu@logos.t.u-tokyo.ac.jp

本論文の構成を以下に述べる。第2章で関連研究の紹介、第3章で本研究の提案手法の説明を行う。第4章で評価の結果を示し、第5章で考察、第6章でまとめと今後の課題について述べる。

2. 関連研究

2.1 教師あり学習

教師あり学習とは、機械学習の手法の一つであり、正解ラベルがつけられたデータを教師データとして利用することで、教師データに含まれていない未知の入力に対しても正答率が高くなるように学習を行う手法である。将棋プログラム Bonanza [2] が成功を収めて以来、将棋の評価関数の学習は教師あり学習で行われるようになってきている。教師データに含まれていないデータに対する性能を汎化性能と言う。教師あり学習では教師データに対して過剰に適合してしまい、未知のデータに対する性能が低下してしまう過学習が起こることがある。そのため学習後は教師データとは異なるデータを用いて汎化性能を評価する。

2.1.1 パーセプトロン

パーセプトロンは教師あり学習において重要な手法の一つである [3]。正例と負例の二値がラベル付けされた教師データから正例・負例を分類するモデルを学習する場合を考える。教師データの例から作成された教師ベクトル x に対しての出力 y は重みベクトル w を用いて以下のように表される。

$$y = f(w \cdot x) \quad (1)$$

ここで $f(x)$ は以下の式で表される。

$$f(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases} \quad (2)$$

出力 y が例についたラベル t と異なっていた場合、重みベクトルを以下のように更新する。

$$w \leftarrow w + c(t - y)x \quad (3)$$

c は学習率と呼ばれる、小さな正の値が用いられる。

パーセプトロンではこの更新を繰り返すことでデータを分類する重みベクトルを学習する。パーセプトロンは正例と負例が線形分離可能である場合は必ず収束し、それらを分離する超平面を見つけることが可能である。

しかし、パーセプトロンはデータが線形分離可能でない場合収束しない。その欠点を改良したものとして平均化パーセプトロン [4] がある。平均化パーセプトロンでは、通常のパーセプトロンと同じ学習を行いつつ過去の重みベクトルの和と重みベクトルの更新回数を記憶しておく。全ての入力データを読みこんだら過去の重みベクトルの平均を計算し、それを最終的な重みベクトルとする。こうすることにより、収束性の問題を解決し、過学習を防ぐことができる。

また、正解データが誤っていた場合だけでなく、 $w \cdot x$ が一定の値以下であったときにも更新を行うことにより、汎化性能を高める手法がある [5]。これをマージンパーセプトロンと言う。

2.1.2 将棋の評価関数の学習

将棋の評価関数の学習において、教師あり学習はよい結果を示している。

例えば Bonanza [2] ではボナンザメソッドと呼ばれる手法が用いられている。これはチェスで用いられていた comparison training [6] をベースとした手法であり、プロの棋譜について、それぞれの局面を訓練例、指された手をラベルと考え、プロが指した手と異なる手を指す回数が少なくなるように学習を行う。ある局面 x から学習に利用する入力ベクトルを作成する際には、その局面 x に対する指し手 t の後の最善応手手順を探索し、得られた末端局面を用いる。棋譜の指し手との不一致率を示す関数に、駒の価値が定数倍変化したような別解を除くための項や過学習を防ぐための項を足し合わせたものを目的関数とし、それを最急降下法により最小化することで評価関数を作成している。

また激指 [7] の学習アルゴリズムは comparison training [6] と平均化マージンパーセプトロンをベースとした手法が用いられている [8]。この手法ではボナンザメソッドと同様、ある局面 x に対する指し手 t のあとの最善応手手順を探索し、得られた末端局面を利用した学習を行う。その局面に対する評価値を返す関数を $f(x, t)$ とし、全ての合法手の集合を T とする。まず棋譜の指し手が他の指し手と比べて相対的に高く評価されているかどうかを確認し、棋譜の指し手より評価値が高い、もしくはある margin を持って評価値が近いものを発見する。具体的には局面 x における棋譜の指し手 t_k に対して

$$T' = \{t \in T | f(x, t) + \text{margin} > f(x, t_k)\} \quad (4)$$

となる指し手の集合 T' を求める。 T' に含まれている指し手は棋譜の指し手と比較して評価値が十分小さくはないということになる。式 (4) の margin は棋譜の指し手を他の指し手と比べてどれだけ評価するかを決める閾値であり、激指では終盤になるほど大きくなるような値を用いている。次に全ての $t \in T'$ の特徴ベクトルについて t_k の特徴ベクトルとの差の和を取り、その平均を重みベクトルに加える。それを全ての局面に対して繰り返し、最後に学習途中に現れた重みベクトルの平均を取り、それを最終的な重みベクトルとしている。

2.2 能動学習

能動学習とは、機械学習における手法の一つであり、少ないラベル付けで高い正解率を得ることを目的としている [9]。ラベルのない訓練例のうち、どの訓練例にラベル

をつけるべきかを決定してからラベル付けを行う手法である。能動学習は、ラベル付けのされていないデータはたくさんあるものの、ラベル付けにコストがかかる問題に適している。どの例にラベル付けを行うかの基準については多くの方針がある。

2.2.1 Uncertainty sampling

Uncertainty sampling は最も単純で広く使われている選択方針であり、最も曖昧な例を選択するという方針で選択を行う [10]。Uncertainty sampling の手法としては、アルゴリズム A において学習に最も有効に働くと考えられる例を x_A^* 、モデル θ において例 x がラベル y となる確率を $P_\theta(y|x)$ とした場合、主に次の 3 つの手法が挙げられる。

- Least confident

「最も確率の高いラベルになる確率」が最も低い例を選ぶ。 \hat{y} をモデル θ において $P_\theta(y|x)$ を最大化するラベルとすると、

$$x_{LC}^* = \operatorname{argmax}_x \{1 - P_\theta(\hat{y}|x)\} \quad (5)$$

となる例 x を選ぶ。

- Margin sampling

「最も確率の高いラベルになる確率」と「2 番目に確率の高いラベルになる確率」の差が最も小さな例を選ぶ。 \hat{y}_i を i 番目になる確率が高いラベルとすると、

$$x_M^* = \operatorname{argmax}_x \{P_\theta(\hat{y}_1|x) - P_\theta(\hat{y}_2|x)\} \quad (6)$$

となる例 x を選ぶ。

- Entropy

予測されるラベルの分布のエントロピーが最も大きな例を選ぶ。つまり、

$$x_E^* = \operatorname{argmax}_x \left\{ - \sum_i P_\theta(y_i|x) \log P_\theta(y_i|x) \right\} \quad (7)$$

となる例 x を選ぶ。能動学習を行った後に最終的に出力するのは最も確率の高いラベルのみである場合、できるだけ多くのラベルについて検討するのは必ずしも得策であるとは限らない。そのため、Entropy を考慮するよりも単純に Least confident や Margin sampling を用いて選択したほうがよいことも多い。

2.2.2 Query-by-committee

Query-by-committee (QBC) は複数の識別器を用いて、各識別器の判断が分かれる例を採用するという手法である [11]。これを実装するためには「判断基準の異なる複数の識別器」と「識別器の判断の不一致度を測る指標」の二つが必要となる。不一致度の指標としては Entropy などが用いられる。

2.2.3 Density-weighted methods

Density-weighted methods は、他の例との類似度が低い孤立した例よりも類似例が多い例を採用するという方針で

ある [12]。Uncertainty sampling や QBC といった方針にこれを加えることにより、類似例のラベルについても推測しやすくなるような例を優先的に選択できる。Density-weighted methods の一つである Information density では以下のように定式化される。

$$x_{ID}^* = \operatorname{argmax}_x \phi_A(x) \times \left(\frac{1}{U} \sum_{u=1}^U \operatorname{sim}(x, x^{(u)}) \right)^\beta \quad (8)$$

ここで $\phi_A(x)$ はある基準となる選択方法 A による選択されやすさであり、 β は類似度の重要度を調整するための定数、 U は例の数、 $\operatorname{sim}(x, x^{(u)})$ は例 x と u 番目の例 $x^{(u)}$ の類似度を測る関数である。 $\frac{1}{U} \sum_{u=1}^U \operatorname{sim}(x, x^{(u)})$ は類似例が多いほど大きな値を示すため、孤立した例が選ばれにくくなる。

3. 提案手法

3.1 局面の選択

本稿では教師データに追加すべき「学習に有効な局面」を選択する手法を提案する。ラベル付けのされていない局面に対して、教師データとなる指し手の確率分布を求めるのは容易ではなく、2.2.1 章で説明したような手法をそのまま用いることは難しい。そのため、確率分布の代わりとして既存の棋譜で学習した評価関数で浅い探索を行ったときの評価値を用いて学習に使う局面を選択する。これにより、その評価関数では判断しにくい局面を選択することができる。将棋は多くの局面において価値の低い合法手が多数存在するという特質がある。このことから考えて、評価値が上位である数個の指し手についてのみ考慮し選択を行う方が、全ての指し手の評価値を考慮して選択を行うよりもよい結果が得られると考えられる。

2.2.1 章で述べた Uncertainty sampling を選択方針とした、以下のような選択基準を提案する。局面 x において指し手 t の評価値を $f(x, t)$ 、 i 番目に高い評価値を持つ指し手を t_i とする。以下の選択基準を順に difference、variance、count、sum とする。

difference 最善手の評価値と次善手の評価値の差が小さい局面

$$x_1^* = \operatorname{argmin}_x \{f(x, t_1) - f(x, t_2)\} \quad (9)$$

variance 上位 3 つの指し手の評価値の分散が小さい局面

$$x_2^* = \operatorname{argmin}_x V(f(x, t_1), f(x, t_2), f(x, t_3)) \quad (10)$$

count 最善手の評価値との差が一定以下の指し手の数が多い局面

$$x_3^* = \operatorname{argmax}_x \operatorname{Count} \{ t \in T | f(x, t) + \text{margin} > f(x, t_1), t \neq t_1 \} \quad (11)$$

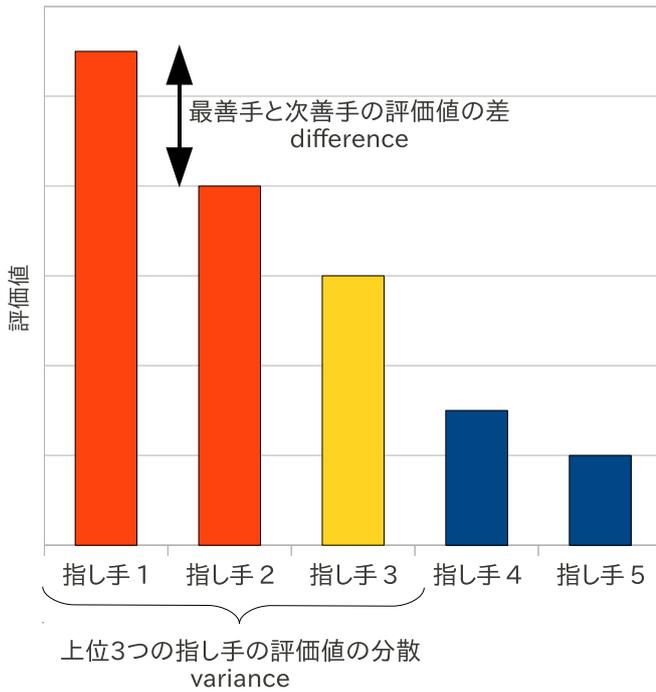


図 1 difference と variance の選択基準

sum count で用いた条件を満たす指し手の $(f(x, t) + margin - f(x, t_1))$ の合計が多い局面

$$x_4^* = \operatorname{argmax}_x \sum_{t \in T'} \{ f(x, t) + margin - f(x, t_1) \} \quad (12)$$

$$T' = \{ t \in T | f(x, t) + margin > f(x, t_1), t \neq t_1 \} \quad (13)$$

difference は 2.2.1 章で述べた Margin sampling を将棋に応用したものであり、variance は、difference を上位 3 つの指し手の評価値に拡大したものである。

count と sum は、激指の学習のアルゴリズムから、学習時に式 4 を満たす指し手が多い局面のほうが学習に有効なのではないかと推測し、そのような特徴を持つ局面が選択できるような基準を採用した。それぞれの選択基準を図 2 に示す。学習時と選択時には

- 学習時は学習途中の評価関数を用いて評価値を計算しているが、選択時はステップ 1 で作成した評価関数を用いている。
- 学習時は棋譜の指し手との評価値の差を利用しているが、選択時は探索した結果最も評価値の高い手との評価値の差を利用している。

などの違いが存在するが、学習時の式 4 を満たす指し手の数を推定するのは難しいので、選択時の評価関数と最善手を用いて、学習時に式 4 を満たす指し手の数が多い局面を推定している。

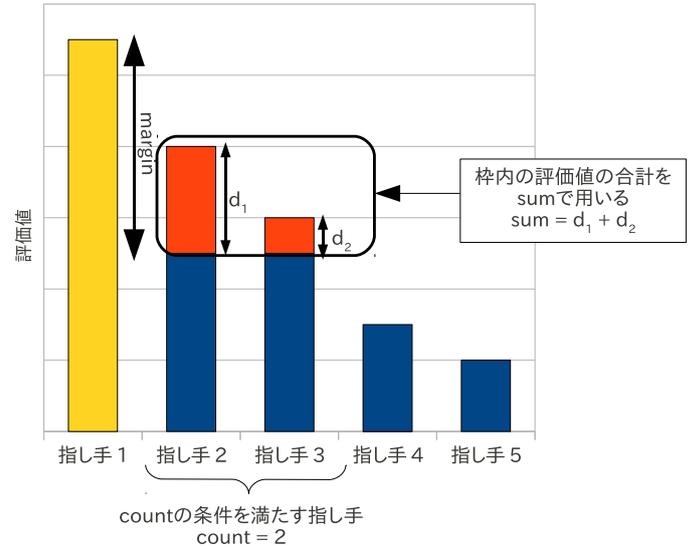


図 2 count と sum の選択基準

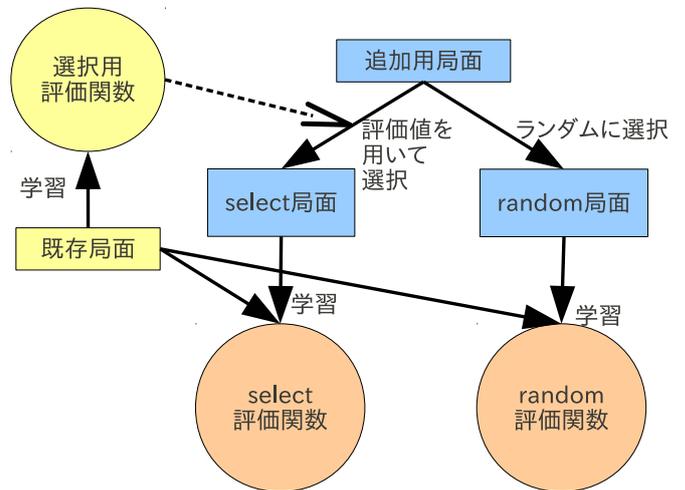


図 3 評価手順

3.2 教師データとしての有用性の評価

提案手法を用いて選択された局面が「学習に有効な局面」であるかどうかを評価する方法として、既存の教師データに、選択された局面を追加して学習を行う。評価手順を図 3 及び以下に示す。

- (1) プロ棋士の棋譜を利用して、学習に用いる局面を選択する際に用いるための評価関数を作成する。
- (2) ステップ 1 で利用した棋譜とは別に追加局面としてプロ棋士の棋譜を用意し、学習に有効であると考えられる局面を選ぶ。
- (3) 選び出された局面とステップ 1 で利用した局面で学習した評価関数 (select) と、ステップ 2 の棋譜からランダムに選ばれた局面とステップ 1 で利用した局面で学習した評価関数 (random) を作成する。
- (4) select と random の評価関数の精度を比較する。

4. 評価

4.1 評価設定

評価には将棋プログラム「激指」[7]を利用した。

3.1章で述べた4つの選択基準を実装し、局面の選択を行った。選択時の評価値としては、ステップ1で作成した評価関数の深さ6での探索結果を利用した。

選択した局面を利用し、3.2章で述べた評価方法を用いて、評価関数の学習を行った。既存の教師データは約50,000局面、追加用の局面は約1,000,000局面、選択する局面数は50,000局面とした。

4.2 評価関数の精度測定

以下の2つの評価方法を用いて、能動学習の効果を計測した。

4.2.1 勝率

randomとselectを対戦させ、その勝率を計測した。対戦は次の2つの設定で行った。

学習に利用していないプロ棋士の棋譜の24手目まで指された1,000局面を用意し、それを初期局面とし先手と後手を入れ替えて2,000対戦を行った。探索の深さは6で固定し、200手を超えた対戦は無効とした。また、探索を深さで打ちきらずに探索時間を1手につき1秒に固定して対戦を行った。対戦回数は探索の深さ6での対戦で利用した開始局面のうち800局面を利用し、先手後手を入れ替えて1,600対戦とした。探索の深さ6で打ち切る場合は1手につき約0.001秒消費していたため、約1,000倍の時間をかけて探索を行うことになる。

4.2.2 一致率

学習や対戦に利用していないプロ棋士の棋譜約100,000局面を利用し、プロ棋士の指し手との一致率を計測した。探索の深さは6で固定した。

4.3 評価結果

評価結果を表1に示す。プロの棋譜の指し手との一致率は、全ての評価関数で42.6%から42.9%までの間に収まっており、能動学習を用いても殆ど違いが見られなかった。最終的に勝利に貢献する評価関数を作成することが目的としているため、プロの棋譜との一致率を用いて評価関数の精度を評価するのは適切ではないと考えられる。全体として一致率が高いと勝率が低くなるという傾向が見られるが、この傾向が有意であるかどうかは更なる検証が必要と

なる。

勝率は選択基準によって差が生じた。特にcountの選択基準で最も勝ち越しており、1手1秒の対戦では二項検定で5%水準で有意に勝ち越した。将棋の評価関数の学習においてうまく選択基準を調整することで、能動学習が効果を発揮すると言える。今回の評価では指し手の評価値のみを用いたが、うまく選択基準を調節することでさらに高い効果を発揮できると考えられる。

似たような選択基準であるdifferenceとvariance、countとsumの間でも、勝率には大きな差が生じている。varianceでは上位3つの指し手の評価値に近いデータが選ばれやすくなるものの、上位2つの指し手の評価値に近いものの3番目の指し手の評価値との間に大きな差があるような場合は選択されないため、これがdifferenceとvarianceの結果の差に影響したのではないかと考えられる。sumでは、 $f(x, t) + margin - f(x, t_1)$ の値が大きな指し手ほど、学習時に式4の条件を満たしやすくと推測したが、この絶対的な値に意味はなく、この推測が間違っていたため、countとsumにの結果に差が生じたのではないかと考えられる。なぜこのような差が生じたか、検証を行っていく必要がある。

5. 考察

5.1 選択方針の再考

今回の評価ではUncertainty samplingを選択方針とした。Uncertainty sampling以外にも将棋の評価関数の能動学習に利用可能であると考えられる選択方針は多々存在し、その有用性を確認するべきである。

2.2.2章で述べたQuery-by-committeeは、複数の将棋プログラムを利用すれば実装可能である。学習や選択にかかる計算のコストがそれだけ増えてしまうという欠点があるが、異なる将棋プログラムであれば並列化することは容易であり、実行時間はUncertainty samplingなどと大きく差がつくことはないと考えられる。

2.2.3章で述べたDensity-weighted methodsは、将棋の場合少しの局面の違いで指すべき手が全く異なっていることは多く、「似たような指し手を選ぶ局面」をどのようにして判断するかが課題になる。しかし、局面の類似度を計算する関数を作成できれば、選択元となる局面を棋譜から生成することにより、実戦に出てきやすい局面を重点的に学習でき、選択方針として有用なのではないかと考えられる。

表1 能動学習による棋譜の選択の結果

	random	difference	variance	count	sum
一致率	42.75%	42.68%	42.84%	42.63%	42.63%
勝率	深さ6 (50.0%)	51.05%	47.99%	52.86%	50.56%
	1秒 (50.0%)	49.97%	48.50%	53.61%	50.64%

5.2 選択基準の追加

今回の評価では指し手の評価値のみを選択基準として用いたが、指し手の評価値以外にも局面の選択に利用できると思われる特徴は存在する。例えば、局面の進行度によって教師データとしての重要度に差があることが分かっており [13]、進行度を選択の際に考慮に入れることでより優れた選択ができると考えられる。他に選択基準として利用できる特徴としては、

- 局面の進行度
- 選択元における同一局面の出現頻度
- 既存の教師データにおける同一局面の出現頻度

などが考えられる。

5.3 教師情報の自己生成

将棋の評価関数の学習において能動学習が有用であることは今回の評価で明らかにできた。今回は全てプロの棋譜を用いたが、プロの棋譜から学習するという手法は、プロ棋士の指し手が好手であるという前提に基づいているため、コンピュータの棋力がプロ棋士を大きく上回るようになるとうまく効果を発揮できなくなる可能性がある。他にもプロ棋士による既存の棋譜は周知の定跡から発展した形のものが多いとあり、新しい戦型に対する棋譜が不足しているといった問題も挙げられる。

これらの問題を解決するためにコンピュータが深く探索して選んだ指し手を教師データに追加することを考える。現在コンピュータ将棋の棋力はプロ棋士に匹敵するレベルに達していると考えられる [14] ので、教師データに利用可能であると思われる。教師情報の自己生成を行うことができれば追加局面に対するラベル付けのコストをさらに削減することが可能になる。先行研究ではコンピュータの深い探索を用いて教師データとして利用可能な指し手を生成するために 1 局面につき 120 秒程度の時間をかけて探索している [1]。一方局面の選択時に消費した探索時間は 1 局面あたり約 0.007 秒であり、選択にかかる時間と指し手の生成にかかる時間には大きな差がある。このため学習に有効に働く局面の選択を行ってから、選択された局面にのみ深い探索を行うことで、よい結果が得られるのではないかと考えられる。

ラベルのついていないデータに深い探索を用いてラベル付けをして新たな評価関数を作成する手順を図 4 及び以下で説明する。

- (1) 既存の教師データを利用して、局面の選択を行うための評価関数を学習する。
- (2) 大量の局面から学習に有効であると考えられる局面を選択する。
- (3) 選び出された局面に対して深い探索を行い、探索結果を用いて局面に対する指し手を決定する。
- (4) 既存の教師データとステップ 3 で指し手を決めた局面

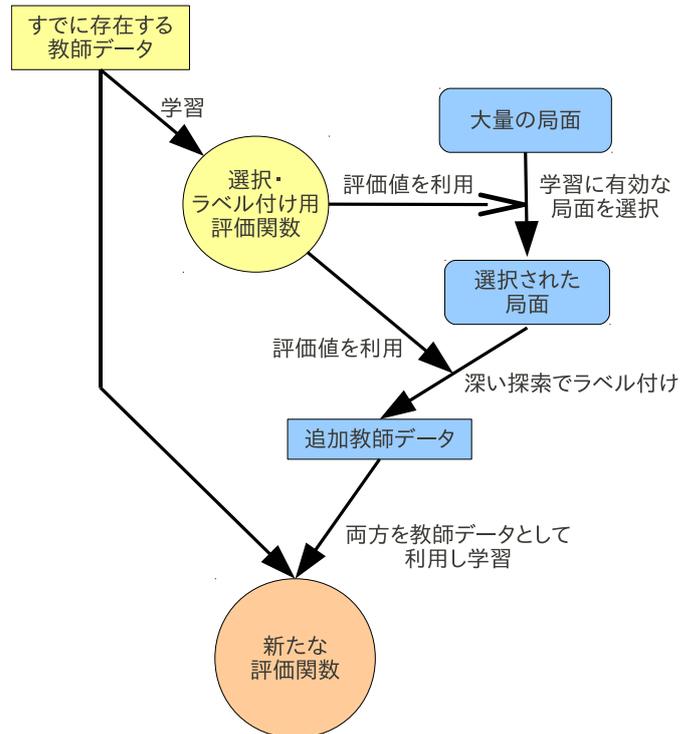


図 4 自己生成を行う学習手順

とともに教師データとして用いて学習を行い、新たな評価関数を作成する。

- (5) ステップ 2 に戻り、この手順を繰り返す。

選択元となるラベル付けの行われていない局面は、指し手の強さが保証されていない棋譜を用いることにより、大量に入手が可能である。具体的には、アマチュア棋士による棋譜や浅い探索による自己対戦の棋譜などが利用可能である。また、学習に有効に働く局面を自己生成するという手法も考えられる。

6. おわりに

本研究では、将棋の評価関数の学習に利用できる教師データの数が限られている問題を解決するために、教師データを増やす際にかかるラベル付けのコストを削減することを目的に、能動学習を用いて教師データに追加する局面を選択する手法を提案した

教師データにランダムに局面を選択して追加した場合と学習に有効に働くと考えられる局面を選択して追加した場合で比較を行った。それぞれを教師データとして用いて学習した評価関数を用いて行った対戦では、最善手との評価値の差が一定以下の評価値を持つ指し手が多い局面を選択した場合、後者が統計的に有意に勝ち越すという結果を得ることができた。このことから、将棋の評価関数の学習において、能動学習を利用することにより少ない教師データで効率のよい学習が可能であると言える。

今後は、今回評価できなかった選択方針についても実装して評価を行っていき、また、選び出された局面に対して

教師データの自己生成を行い、自己生成された教師データをプロの棋譜に追加して学習するという手法により、既存の棋譜のみで学習した評価関数よりも精度が高い評価関数を作成することを目指したい。

報：4. 第 23 回世界コンピュータ将棋選手権速報，情報処理，Vol. 54, No. 9, pp. 925–928 (2013).

参考文献

- [1] Ura, A., Miwa, M., Tsuruoka, Y. and Chikayama, T.: Comparison Training of Shogi Evaluation Functions with Self-Generated Training Positions and Moves, *The 8th International Conference on Computers and Games (CG 2013)* (2013).
- [2] 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 2006，pp. 78–83 (2006).
- [3] Rosenblatt, F.: The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, *Psychological Review*, Vol. 65, No. 6, pp. 386–408 (1958).
- [4] Freund, Y. and Schapire, R. E.: Large Margin Classification Using the Perceptron Algorithm, *Mach. Learn.*, Vol. 37, No. 3, pp. 277–296 (online), DOI: 10.1023/A:1007662407062 (1999).
- [5] 原 一之，岡田真人：マージンを用いた単純パーセプトロン学習法のオンラインラーニングの理論，電子情報通信学会技術研究報告. NC, ニューロコンピューティング，Vol. 101, No. 534, pp. 51–57 (オンライン)，入手先 (<http://ci.nii.ac.jp/naid/110003233965/>) (2001).
- [6] Tesauro, G.: Comparison training of chess evaluation functions, *Machines that learn to play games* (Fürnkranz, J. and Kubat, M., eds.), Nova Science Publishers, Inc., Commack, NY, USA, pp. 117–130 (online), available from (<http://dl.acm.org/citation.cfm?id=644391.644397>) (2001).
- [7] 将棋プログラム「激指」のページ，入手先 (<http://www.logos.t.u-tokyo.ac.jp/gekisashi/>).
- [8] 矢野友貴，三輪 誠，横山大作，近山 隆：ゲーム構成要素を組み合わせた特徴の最適化，ゲームプログラミングワークショップ 2010，pp. 15–22 (2010).
- [9] Settles, B.: Active Learning Literature Survey, Technical Report 1648, University of Wisconsin–Madison (2009).
- [10] Lewis, D. D. and Gale, W. A.: A sequential algorithm for training text classifiers, *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '94, New York, NY, USA, Springer-Verlag New York, Inc., pp. 3–12 (online), available from (<http://dl.acm.org/citation.cfm?id=188490.188495>) (1994).
- [11] Seung, H. S., Opper, M. and Sompolinsky, H.: Query by committee, *Proceedings of the fifth annual workshop on Computational learning theory*, COLT '92, New York, NY, USA, ACM, pp. 287–294 (online), DOI: 10.1145/130385.130417 (1992).
- [12] Settles, B. and Craven, M.: An analysis of active learning strategies for sequence labeling tasks, *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, Stroudsburg, PA, USA, Association for Computational Linguistics, pp. 1070–1079 (online), available from (<http://dl.acm.org/citation.cfm?id=1613715.1613855>) (2008).
- [13] 佐藤佳州，高橋大介：大規模な対局に基づいた教師データの重要度の学習，ゲームプログラミングワークショップ 2012，pp. 22–29 (2012).
- [14] 瀧澤武信：現役プロ棋士に勝ち越したコンピュータ将棋 第 2 回電王戦，第 23 回世界コンピュータ将棋選手権速