局面情報からの探索信頼性の推定による将棋の ProbCut の性能向上

亀甲 博貴[1,a)]　浦 晃[2]　三輪 誠[3]　鶴岡 慶雅[2]　近山 隆[2]

概要：
将棋などのコンピュータプレイヤは，局面の形勢判断を行う評価関数を用いてゲーム木探索を行っている．しかし評価関数が返す値はあくまで注目する局面を点として見たときの価値でありその価値がどの程度安定しているものであるか分からない．その局面からどれだけ形勢が変化しうるのかを検討することはその探索結果が信頼できるものであるかを考える上で重要である．本研究では浅い探索で得られる評価値と深い探索で得られる評価値の間の相関を調べる．また浅い探索の信頼性が浅い探索で得られる評価値と深い探索で得られる評価値の差の分布で表現できると仮定して，注目局面の特徴情報から深浅差の標準偏差を推定する手法を提案する．深浅差が小さいと予測される局面は浅い探索のみを行えば深い探索を省略してもその危険性が少ないと考えられるので，そのような局面について ProbCut を行うことで将棋プログラムの棋力向上を図る．将棋への pProbCut への有効性は確認できたが，提案手法による ProbCut の有意な性能向上は見られなかった．

Improving the Performance of ProbCut by Using Positional Features in Shogi

Hirotaka KAMEKO[1,a)]　Akira URA[2]　Makoto MIWA[3]　Yoshimasa TSURUOKA[2]
Takashi CHIKAYAMA[2]

Abstract:
Computer shogi players usually use evaluation function in game-tree search. However, the values returned by evaluation functions are point estimates of the values of the positions and do not convey information about how reliable they are. It is important to estimate their level of reliability by quantifying how much those values may change. In this paper, we investigate the distribution of gaps between shallow and deep evaluation values and propose two methods to estimate it. We also try to improve performance of ProbCut using the variance of distribution. ProbCut is effective in Shogi, but proposed method couldn't improve performance of ProbCut.

1. はじめに

アルゴリズムの面で，将棋プログラムの棋力が向上する要因の 1 つに，探索効率の向上があげられる．ゲーム木探索を用いるゲームでは，一般にゲーム木を深く読む方がよい手を選択しやすいことが経験的に知られている．ゲームのルールとして各プレイヤには持ち時間が存在し，またマシンスペックにも限界がある．そのため Bitboard の導入とその応用 [1] などの，時間あたりの探索ノード数を増やす工夫とともに，限られた探索ノード数をより有効に使う工夫が棋力向上につながると考えられている．ゆえに探索する価値のないノードの探索を極力回避して，指し手の選択を左右しうるノードをより多く探索する工夫は，プログラムの棋力向上を図る上で欠かすことができない．そのような

[1] 東京大学工学部電子情報工学科
Engineering Department, 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学科
School of Computer Science, The University of Manchester
[a)] kameko@logos.t.u-tokyo.ac.jp

工夫として Probcut [2]，Null Move Forward Pruning [3]，Futility Pruning [4] などが挙げられる．

これらの工夫の一部は，深い探索が不要であることを浅い探索の結果から予測している．浅い探索で得られる評価値と深い探索で得られる評価値の間の相関を，本論文では浅い探索の信頼性と表現する．相関が強い局面は浅い探索の信頼性が高いとする．浅い探索の信頼性が強ければ，前述の枝刈りは浅い探索の結果を用いて枝刈りの可否を決定できる．本論文では浅い探索で得られる評価値と深い探索で得られる評価値の差を「深浅差」と表現し，

$$\text{深浅差} = (\text{深い探索で得た評価値}) - (\text{浅い探索で得た評価値})$$

と定義する．この深浅差の分布が浅い探索の信頼性の指標に使われることが多い．

そこで本研究では，局面の様々な特徴に注目して浅い探索の信頼性を推定する手法と，これを用いての ProbCut を提案する．注目するノードの特徴から，その局面に対して深浅差の分布を標準偏差として推定することで浅い探索の信頼性とする．1 つ目の手法は標準偏差を各進行度ごとに計算して線形回帰することで，特徴の有無と局面の進行度から標準偏差を計算する．2 つ目の手法は特徴の有無と局面の進行度を用いて線形 SVR（Support Vector Regression） [5] により深浅差の絶対値を推定する．

進行度とはその局面がゲームの展開の中のどの位置にあるかを示す値で，盤面上の駒の位置や玉への駒の効きなどから計算される．開始局面の進行度は 0 で，0 から 127 の 128 値を取る．終局に近いほど大きな値を取る．一方で局面の特徴は，その全てがゲームの展開と強い相関を持つわけではない．そのようなゲームの展開との相関が弱い特徴を進行度と組み合わせることで，浅い探索の信頼性をより正しく推定する．

提案手法により決定された特徴と浅い探索の結果をもとに，深い探索から得られる評価値が取りうる範囲を予測し，ProbCut の可否を決定する．これにより不要な深い探索を削減し，限られた時間で指し手に影響を及ぼしうる部分をより深く探索できることが期待される．

評価関数が返す値がどの程度信用できるものであるかを示す指標の 1 つとして，浅い探索の信頼性を提案する．これにより一般的な評価関数では表現できなかった，評価値のゆらぎを見積もることができ，評価値の不安定性が改善できることが期待される．

本論文の以下の構成は以下の通りである．2 章で関連研究を紹介し，3 章で深浅差の分布について述べる．4 章で提案手法の詳細について述べ，続く 5 章で提案手法を評価するために行なった実験とその結果を示す．最後に 6 章でまとめと今後の課題について述べる．

図 1　浅い探索と深い探索で結果が大きく異なる局面（shogipic.jp を用いて作成．）

2. 関連研究

2.1 ProbCut

$\alpha\beta$ 探索は枝刈りが結果に影響を及ぼさない厳密な枝刈りで，これを後ろ向き枝刈りと呼ぶ．一方でより多くの枝刈りを行おうとする場合には，$\alpha\beta$ 探索と同様の結果を返す厳密な枝刈りではなくなる．このように枝刈りを行うことによって結果が変わりうる枝刈りを前向き枝刈りと呼ぶが，ProbCut [2] はそのような前向き枝刈りの 1 つで，浅い探索に高い信頼性があることを利用して枝刈りを行う．

将棋での ProbCut の具体例を以下に述べる．ある残り深さ 8 のノードにたどり着いたとき，一旦深さ 4 で探索を行う．その結果がノードの $\alpha\beta$Window から著しく外れていた場合は，深さ 8 で探索を行ったとしてもおそらく枝刈りされるだろうという予測のもとで探索を打ち切る．枝刈りの可否の基準として一定のマージンを設定し，浅い探索の結果が $\alpha - margin$ より小さい場合は α カットを，$\beta + margin$ より大きい場合は β カットを行う．

この枝刈りは深浅差が一定値を下回ることを期待しているが，例えば図 1 のような局面を将棋プログラム「激指」*1 の評価関数に与えると，深さ 4 での探索では評価値 -673 であったものが深さ 8 での探索では評価値 380 となり，値が著しくずれてしまっている （歩 1 枚の基本価値は 100 である）．統計データ全体から求められる深浅差の標準偏差は約 300 であり，これをマージンとして枝刈りを行うと，深い探索の評価値が取るであろうと予測される範囲から大きく外れてしまいかねない．このような場合は，ProbCut によって本来返すべきノードを枝刈りしてしまう危険性がある．しかし $\alpha\beta$ 探索自体が探索をある程度で打ち切る以上，特に中盤以前の局面では $\alpha\beta$ 探索が真に勝利

*1 http://www.logos.ic.i.u-tokyo.ac.jp/ gekisashi/ （2012 年 10 月 2 日現在）

へつながる手を返すとは限らない．そのため厳密に $\alpha\beta$ 探索と同じ結果であることを求める必要性は低く，ある程度結果が変わることについては許容できることが期待される．

ProbCut はオセロなどのゲームにおいてその有効性が示されている．将棋においても浅い探索の結果に高い信頼性が見られ，静止探索*2において ProbCut を将棋に導入することの有効性が示されている [6].

また ProbCut を複数深さ間で実行する Multi ProbCut という手法が考案され，それぞれの深さについて適切なパラメタを事前に用意することでオセロで ProbCut や通常探索を用いた相手に有意に勝ち越す結果を得ている [7].

2.2 Null Window 探索

ProbCut を行うにあたり，枝刈りが可能であるかどうかの判定には浅い探索を行う．しかし枝刈りが生じない場合は深い探索も行わなければならないため二度手間となり，探索ノード数は逆に増えてしまう．浅い探索での余計な探索量を削減するための 1 つの手法として，Null Window 探索がある．

$\alpha\beta$ 探索はそのノードの評価値が（1）α 以下であるときは α（2）β 以上であるときは β（3）$\alpha < x < \beta$ となる値 x であるときはその値 x，の 3 通りの結果を返す．（3）の場合以外は枝刈りが起こっており正確な値は判明しないが，それぞれ α 以下であるまたは β 以上であるという情報を得ることはできる．そこであるノードの正確な評価値に興味がなく，評価値が α_0 以下であるかどうかのみを知りたい場合，$\alpha = \alpha_0$，$\beta = \alpha_0 + 1$ とするとその結果から α_0 以下であるか否かが判明する．このとき Window 幅が 1 であるため自分手番ノードの評価値が $alpha_0$ 以下または相手手番ノードの評価値が $alpha_0 + 1$ 以上の値を取った時点で枝刈りが生じる．Null Window 探索はこのように Window 幅を 1 として $\alpha\beta$ 探索を行う手法で，早い段階で枝刈りが生じることから探索結果が注目する値以上であるか否かを高速で得ることができる．しかし通常探索と違い最善手順を得られるとは限らず，浅い探索の結果を用いることはできない．

2.3 その他の枝刈り手法

有力な前向き枝刈り手法の 1 つに，Null Move Forward Pruning [3] がある．将棋はルール上パスができないが，仮にパスをしたとして探索を行う．適切な指し手は指さない場合より常によい状態を得られるという発想に基づいており，パスしたときの評価値が β 値を超えていれば，適切な指し手を選べばより高い評価値を得られるだろうことが予想されるので枝刈りをするという手法である．

別の前向き枝刈り手法に Futility Pruning [4] と呼ばれる手法がある．探索を行い，残り 1 手となったときにそのノードを評価する．そのの評価値が α より一定以上低いまたは β より一定以上高いとき，後 1 手指してもその評価値が α と β の間に入ることはないだろうという予測のもとで枝刈りをする．Futility Pruning はコンピュータ将棋選手権で優勝経験を持ち将棋プログラムに大きな影響を与えた Bonanza で応用され棋力の上昇が示されている [8].

これらの枝刈りが ProbCut による枝刈り部分と共通する場合，浅い探索を不必要に行なってしまう可能性がある．特に Null Move Forward Pruning は ProbCut が起こるのと同じタイミングでも起こりうる枝刈りであるためその影響は大きいことが予想される．

2.4 局面ごとにマージンを変化させる枝刈り手法

前項までで紹介した ProbCut，Null Move Forward Pruning，Futility Pruning といった枝刈り手法をゲームの全体で固定のマージン値を用いて枝刈り判定を行うと，誤った枝刈りが生じかねない．そこで局面の状態によって深浅差の分布が変わることから，マージン値を変動させる手法が考えられてきた．

将棋に ProbCut を導入した例として，局面までの手数を用いて通常探索に ProbCut を適用し，その有効性を示した研究がある [9]．この研究において，2 手おきに評価値の差を取ることで統計的性質を調査し，初期局面から現在局面までの手数ごとに標準偏差を計算すると，それが手数で直線に近似できることを示した．また前述の近似から求まる標準偏差に適切な係数を掛けたものをマージンとして用いた ProbCut が，通常の $\alpha\beta$ 探索に有意に勝ち越すことを示した．

この研究に対し，各局面を序盤・中盤・終盤といった局面の展開に分類してマージンを定める手法が提案された [10]．この研究ではゲームの展開を駒の衝突・駒の成り・寄せ・詰めといったイベントで分類し，各分類ごとに回帰を行なってマージンを決定する．これらはいずれもマージンの決定関数を事前に与える手法である．前述の手数によるマージンの決定より局面の展開を適切に分類できているが，これもゲーム中の流れのみによる分類である．この研究はまた終盤は詰みの読みぬけの恐れから ProbCut の適用が難しいと考察している．

対して対局の途中でマージンを動的に変動させる手法を Futility Pruning に導入して棋力の向上を示した研究がある [11]．これは探索を行う中で仮に枝刈りを行なったとしてその枝刈りが正当なものである確率を実際に数え上げ，探索中に一定数の統計が取れたらその確率を今探索している局面に用いて枝刈りを行う手法である．一定数の統計が取れるまでは枝刈りが行われないため，ある程度以上の計算時間を与える必要があるが，事前にプログラムにゲームに関する知識を与える必要がない点は前述の手法との大き

*2 通常探索ではなく，駒の取り合いなどの手のみを読んで局面を静止させる探索

な相違点である．

これらの先行研究は，いずれも指し手の選び方で強く変動することの少ない情報を用いている．選ぶ指し手に関わらず初期局面から現在局面までの手数は同じで，適切な指し手を選べば同じ手番の探索中に序盤・中盤・終盤といった展開が大きく変動することは少ないと考えられる．また探索中に統計を取る手法は，同じ手番では同じマージンを用いることを前提としている．これらの手法に対し本研究で提案する手法は，同じ手番での探索中に現れる各局面それぞれで安定性が異なるという予測のもと，その安定性の違いを推定することを目標とする．

3. 深浅差の分布

本研究に先立ち，深浅差の実際の分布を調査した．プロ棋士の棋譜に現れる各局面に複数深さで探索を行い，それらの評価値差を見た．本論文では ProbCut を深さ 8 の時点で行うために深さ 4 と深さ 8 の深浅差に注目する．

調査には将棋プログラム「激指」を用いた．プロの棋譜中に現れる各局面に対してそれぞれの深さで探索を行い，評価値と最善手順を得た．

全局面から求めた深浅差の分布を図 2 に示す．回帰により正規分布と 2 重混合正規分布での近似を重ねている．この図から深浅差の分布は 2 重混合正規分布に近いことがわかる．2 重混合正規分布は図の通り単純な正規分布に比して裾野の広い確率分布で，正規分布と比べると信頼区間が狭い．ProbCut は深浅差が一定の区間内にあることを期待しての枝刈りで深浅差がその外側であるときは誤った枝刈りをしてしまう危険性があるが，深浅差の分布を正規分布と仮定して枝刈りを行うと本来期待される確率より誤りの確率が高くなってしまう．

浅い探索と深い探索をそれぞれ深さ 4 と深さ 8 で行うが，実現確率探索 [12] や静止探索といった探索の縮退・延長手法を実装していることからルート局面から最善手順でのリーフ局面までの手数はそれぞれ 4 手，8 手とは限らない．974,055 局面に深さ 4 の探索と深さ 8 の探索をした結果それぞれの手数が 4 手・8 手となった局面が 153,993 局面であったのに対し 1 手・3 手となった局面が 176,656 局面存在する．これらの深浅差の分布を図 3，図 4 に示す．手数が固定であれば深浅差の分布はほぼ正規分布であるといえ，この分布の重ねあわせから混合正規分布が現れているものと考えられる．

浅い探索の最善手順の手数が 4 手のものと 1 手のものを抜き出しての深浅差の分布が図 5 と図 6 である．浅い探索の最善手順の手数が分かると，深浅差の分布がより詳しく推定できることが分かる．

但し ProbCut を行うにあたって浅い探索は Null Window 探索によって行われる．Null Window 探索では最善手順を探索するとは限らず浅い探索で得られる最善手順の手数を

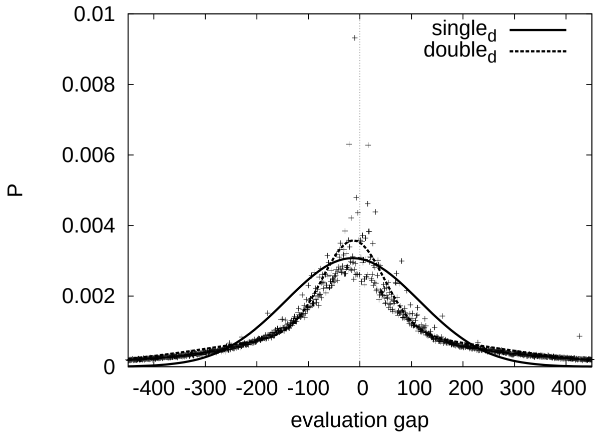

図 2 全体から得られる深浅差の分布

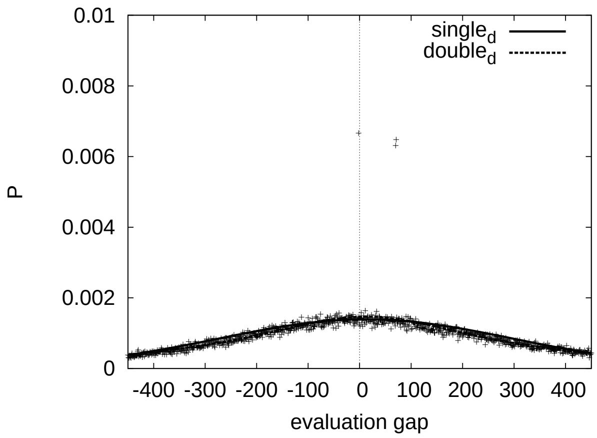

図 3 手数 4 手と 8 手の深浅差の分布

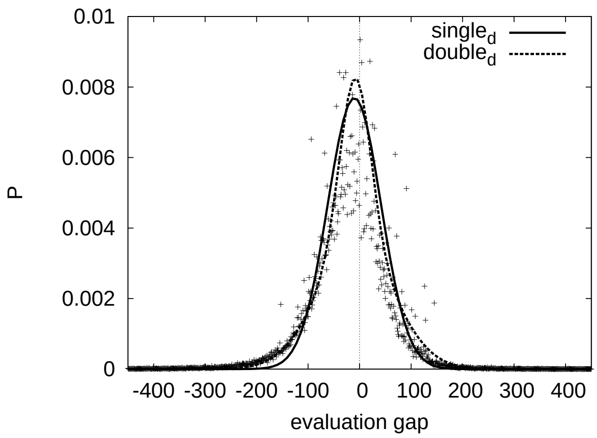

図 4 手数 1 手と 3 手の深浅差の分布

得ることはできない．一方で浅い探索の評価値を通常探索で得ると浅い探索の探索量のために枝刈り効率が落ちてしまう．そのため ProbCut の性能向上を考えるときに深浅差のより適切な分布推定は難しいことが予想される．

深浅差の標準偏差を各進行度ごとに計算してプロットしたのが図 7 である．標準偏差と進行度は直線に近い関係を持つ．回帰により得た直線 $2.90t + 95.0$ を重ねている（t は進行度）．

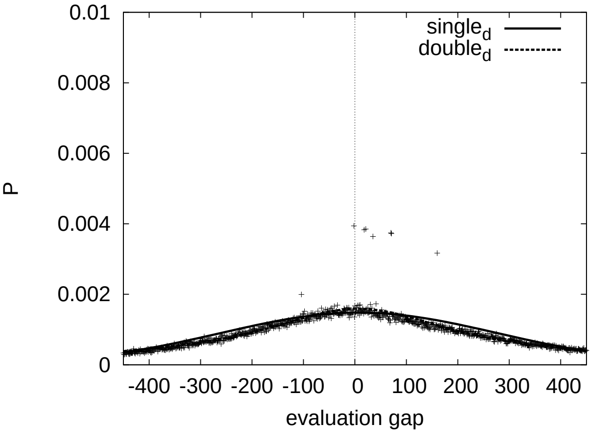

図 5　浅い探索の最善手順の手数 4 手の深浅差の分布

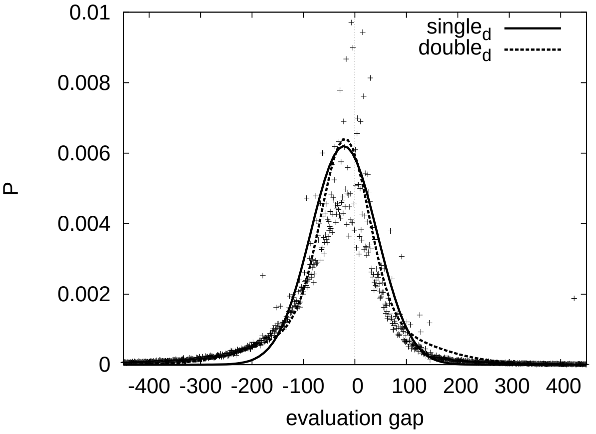

図 6　浅い探索の最善手順の手数 1 手の深浅差の分布

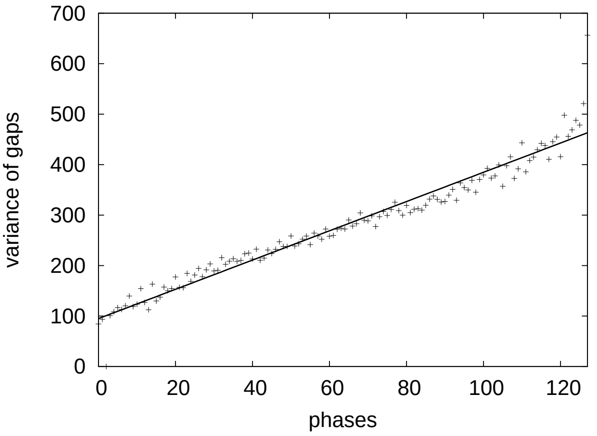

図 7　全体から得られる深浅差と進行度との関係

4. 提案手法

本研究では，ProbCut で用いるマージン値を枝刈りの可否を判定する局面の特徴から決定する 2 つの手法を提案する．浅い探索の結果に高い信頼性があるという仮定のもと，1 つ目の手法は深浅差の分布モデルを複数用意し，注目局面の特徴からその分布モデルに分類する．分類された各モデルについてその標準偏差の値を用いて ProbCut のマージン値を決定する．2 つ目の手法は 1 つ目の手法で用いたものと同じ特徴を用いて線形 SVR により各局面に深浅差の絶対値を与える．また浅い探索の結果の信頼性が低いノードが存在しそれが検出できるならばそのノードについて枝刈りを適用しないことで誤り枝刈り数を減少させる．

4.1　局面の分類に用いる特徴の決定

局面の分類には評価関数に用いられるような駒割などの特徴を用いる．しかし枝刈りの可否を判定する局面ごとにマージン値を決定する必要があり，特徴の数が増えると計算量の増加から単位時間当たりの処理ノード数が減少する．これを回避するため特に影響の大きい特徴のみを抜き出す．

注目する各特徴についてそれぞれ局面を 2 値分類し，特徴を有する局面群と有さない局面群に分ける．これらの局面群と局面全体に対し，特定の進行度間にある局面のみを抜き出してそれぞれの深浅差の標準偏差を求める．局面全体から得られた標準偏差とどちらかの局面群から得られた標準偏差が大きく離れていたものを候補特徴とする．この候補特徴を用いて全進行度において標準偏差を求め，局面全体について全進行度で求めた標準偏差と有意に離れていることが確認できた特徴を分類に用いる特徴として採用する．

4.2　標準偏差の推定手法

深浅差の分布を期待値 $\mu = 0$，分散 σ^2 の正規分布 $\mathcal{N}(\mu = 0, \sigma^2)$ と仮定し，前項で決定した特徴を用いて注目局面に対して深浅差の標準偏差を与える．推定の手法として特徴から局面を分類して各分類ごとに標準偏差を計算する手法と線形 SVR を用いる手法の 2 つを提案する．

4.2.1　局面の分類による標準偏差の推定

各特徴の有無で局面を分類し，各分類ごとに標準偏差を計算する．各進行度ごとに標準偏差を計算し，線形回帰を行うことで各分類ごとに標準偏差を直線の式として得る．但し n 個の特徴の有無について 2^n に分類し更に 128 値を取る進行度で分類すると分類がスパースになることが予測されるため，各特徴に優先順位をつけて $n + 1$ 分類に対して標準偏差を求める．

4.2.2　線形 SVR による標準偏差の推定

深浅差の分散または標準偏差は評価値への特徴の寄与の分散または標準偏差の総和であるという仮定から，深浅差の分散や標準偏差を各特徴の線形和で表現する．

分散の線形モデルの作成には線形 SVR [5] を用いる．各局面を深浅差の期待値 $\mu = 0$，標準偏差 $\sigma = gap$ の分布とみなし，目的変数に深浅差の 2 乗または深浅差の絶対値，説明変数に進行度と特徴を持つときは 1，持たないときは -1 の特徴量を持つ特徴ベクトルからなるデータ群からモデルを作成する．それぞれの値は $[-1, 1]$ に正規化する．

線形 SVR は学習器の 1 つで，$y_i = \mathbf{x}_i\mathbf{w} + b$ を求める．本研究では y_i に深浅差の 2 乗または深浅差の絶対値を，$\mathbf{x}_i$ に注目局面の特徴ベクトルを与えて学習する．

5. 評価

5.1 評価に用いたプログラム

本研究の評価には将棋プログラム「激指」を用いた．激指は実現確率探索を用いたゲーム木探索を行なっており，深さ n 手といったときには厳密には n 手読まず遷移確率の低い指し手については木を縮退している．但し縮退を行わなかった経路の探索手数はおおむね n 手である．またリーフノードでは静止探索を行なっており，静止するまでは木を延長している．残り深さ n といったときは厳密に n 手ではなく，またこの場合の n は整数値とは限らない．

ProbCut の実装は以下の通りである．

- 各ノードにおいて残り深さが 5 未満または 9 以上[*3]なら枝刈りを行わない．
- Window が $\alpha < -30,000$（αPrubCut），$\beta > 30,000$（βProbCut）であればそれぞれの枝刈りを行わない．
- 自玉に王手がかかっていれば枝刈りを行わない．

以上の判定ののち残り深さを 4 減じた上でマージンを定め Null Window 探索を行なった．なお本研究では α カットと β カットで同じマージンを用いた．[*4]

前向き枝刈り手法として Null Move Forward Pruning と Futility Pruning を行なっている．特に Null Move Forward Pruning は探索途中の各ノードで枝刈り判定を行うため ProbCut との競合が予想される．

比較対象は，

- ProbCut を行わない
- マージンを固定した ProbCut
- 進行度のみによってマージンが決定される ProbCut $\sigma = 2.9t + 95.0$（t は進行度）

の 3 つである．マージンや係数を変動させて提案手法との比較を行う．

5.2 標準偏差の推定

推定に用いる特徴として

- 持ち駒に香車がある
- 自飛車と自玉が近い（2 列以内にいる）
- 自角が敵金銀に効いている
- 自角と敵玉が近い（自玉と敵玉それぞれと自各の距離を比較して敵玉の方が近い）
- 自角と 5 五が遠い（角が盤面上にあり，かつ 5 五にない）

の 5 つを得た．これらの特徴を自分が持つ場合と相手が持つ場合で標準偏差の変化に大きな違いが見られなかったため，いずれかのプレイヤが特徴を持つ場合はその特徴を持つ局面として扱った．

この特徴から特徴ベクトルを作り，線形 SVR を用いて深浅差の推定を試みた．約 77 万局面のデータを用いて，学習器として LIBLINEAR を用いてパラメタを変動させて SVR を行いその精度を 10 分割交差検定で計測した．目的変数として深浅差の 2 乗を与えたところ MSE（Mean Squared Error，平均二乗誤差．誤差の 2 乗の平均．）9.00×10^{-3}，二乗相関係数 0.032 であり高い相関が得られなかった．そこで目的変数は深浅差の絶対値として同様に行なったところ MSE5.33×10^{-3}，二乗相関係数 0.154 程度のモデルを得た．しかしモデルの係数を確認するとその実態はほぼ進行度の寄与のみで深浅差が与えられ，その他の特徴の寄与は MSE 以下であった．得られた係数を表 1 に示す．ラベルとして与えた深浅差の絶対値は正規化して $1/10,000$ としているので実際はこの 10,000 倍の値を取る．この結果から線形 SVR を用いた提案手法の有効性を示すことはできなかった．

原因として特徴の少なさとそれぞれの特徴が取れる特徴量の少なさが挙げられる．本研究では深浅差の絶対値 $10,000$ 以下を対象としたが，進行度を考慮しなければこれを 2 値の特徴 5 つの 32 分類で表現しなければならない．そのため各特徴が深浅差に有意な差となって現れることができなかったのだと考えられる．

特徴を用いて 6 分類しての標準偏差を表 2 に示す．なお表の上の特徴ほど優先度が高い．また上 2 つの状態と全体の各進行度での標準偏差を図 8 に示す．図を見る限り，特に進行度が大きい局面では局面の分類が標準偏差に有意な差となって現れているように見える．逆に進行度の小さい局面では各状態間で進行度に大きな差が現れていない．

5.3 プロ棋士の棋譜との一致率と探索ノード数の削減量

ProbCut による探索ノード数の削減量を見るため，プロ棋士の棋譜に現れる 1,000 局面に対して深さ固定での探索を行い指し手の一致率と計算量を確認した．結果を表 3 に示す．

表 1 線形 SVR から得られたモデルの係数

進行度	0.0285
持駒に香車がある	-0.00156
飛車と自玉が近い	0.00132
角が敵金銀に効く	-0.000117
角と相手玉の距離	-0.00164
角と中央との距離	0.00138
定数項	0.0856

[*3] 実現確率探索を用いており，厳密に深さ 8 手のノードで枝刈りを行うことができないため深さ 8 付近で枝刈りを行う．また浅い探索は残り深さを 4 減算することからその上で残り深さが 1 手以上残る必要がある．

[*4] α カットが発生すると判定された場合は詰み探索を行い詰みが見つかったらそれを返す．詰み探索が見つからなかった場合は α カットを行う．β カットが発生すると判定された場合はそのままカットを行う．

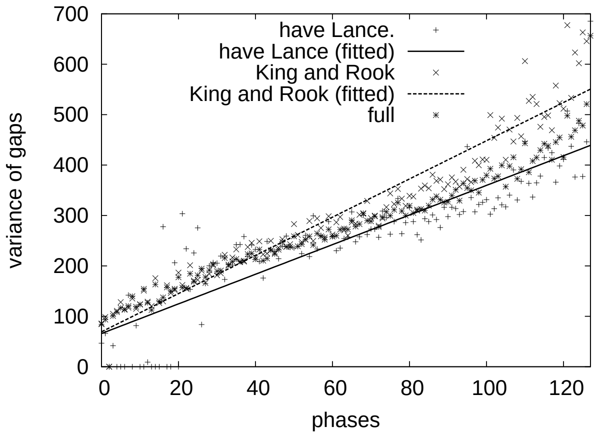

図 8　各状態の標準偏差と回帰直線

ProbCut を行うことにより単位時間当たりの処理ノード数は減少したもののそれを上回る探索ノード数の削減に成功しており実時間での探索時間を大きく短縮できている．

表 2　各状態の標準偏差（t は進行度）

持駒に香車がある	$2.94t + 65.9$
飛車と自玉が近い	$3.80t + 69.3$
角が敵金銀に効く	$3.40t + 92.8$
角と相手玉の距離	$2.52t + 98.7$
角と中央との距離	$3.35t + 76.7$
その他	$3.41t + 25.9$

表 3　探索ノード数の比較

	一致数	時間（s）	ノード数（%）	#/s
ProbCut なし	574	4,294	402M（100）	93,722
固定（100）	552	2,049	147M（36.6）	71,795
固定（300）	564	2,646	205M（51.0）	77,514
固定（800）	569	3,272	303M（75.4）	92,670
進行度（3.0σ）	575	2,626	221M（55.0）	94,476
変動（1.0σ）	558	2,640	209M（52.0）	78,222
変動（2.0σ）	567	3,315	303M（75.4）	91,495
変動（3.0σ）	591	3,701	362M（90.0）	97,725
変動（4.0σ）	578	3,790	389M（96.8）	102,691
固定（300）β	569	3,237	273M（67.9）	84,238
固定（300）α	564	2,930	238M（59.2）	81,246

表 4　探索ノード数の比較（進行度 96 以上）

	一致数	時間（s）	ノード数（%）	#/s
ProbCut なし	658	8,251	941M（100）	114,087
固定（100）	626	3,504	296M（31.5）	84,473
固定（300）	643	4,451	419M（44.5）	94,020
固定（800）	657	4,624	476M（44.5）	102,974
進行度（3.0σ）	641	3,581	338M（35.9）	94,476
変動（1.0σ）	642	5,279	523M（55.6）	99,124
変動（2.0σ）	665	4,870	506M（53.8）	103,913
変動（3.0σ）	670	5,364	593M（63.0）	110,456
変動（4.0σ）	671	5,565	633M（67.3）	113,663

またマージンの値を小さくしても指し手の一致数に大幅な減少は見られず，この結果だけから棋力の変化を確認することはできない．

同様の実験を今度は進行度 96 以上の 1,000 局面に対して行なった．結果を表 4 に示す．このときは固定マージンでは探索時間を削減するためにマージン値を小さくすると一致率が減少した．一方で変動マージンはその係数を小さくしても比較的安全に枝刈りが行われている．

しかしいずれの場合も進行度のみからマージンを決定することで高い枝刈り効率を出しており，この結果からは提案手法の有効性を示すことはできない．

5.4　自己対戦による棋力の比較

ProbCut の有無と各パラメタの違いによる棋力の比較を自己対戦により行なった．プロ棋士の棋譜から 35 手読み，1 手の時間を固定しての対局を行う．200 手（お互いに 100 手ずつ）指して決着がつかなかった場合は引き分けとする．同一局面について先後を入れ替えて対局する．210 局面 420 対戦を行い，引き分けは 0.5 勝 0.5 敗として勝率を計算する．結果を表 5 に示す．

マージンの係数を小さくすると大幅に負け越した．またマージンの係数を大きくしすぎても負け越すようになった．適切な係数を与えることで ProbCut をしない相手に対して有意に勝ち越すことが確認できた．

表 5　自己対戦の結果

プレイヤ	1 手	結果（win-draw-lose）
変動 1.0σ vs ProbCut 無し	3 秒	167-32-221 （0.436）
変動 2.0σ vs ProbCut 無し	3 秒	185-17-218 （0.460）
変動 3.0σ vs ProbCut 無し	3 秒	209-23-188 （0.525）
変動 4.0σ vs ProbCut 無し	3 秒	192-23-205 （0.485）
固定 300 vs ProbCut 無し	3 秒	141-32-243 （0.374）
固定 800 vs ProbCut 無し	3 秒	209-27-184 （0.530）
進行度 3.0σ vs ProbCut 無し	3 秒	198-28-194 （0.505）
進行度 4.0σ vs ProbCut 無し	3 秒	201-26-193 （0.510）
変動 3.0σ vs 固定 500	3 秒	202-21-197 （0.506）
変動 3.0σ vs 固定 800	3 秒	205-20-195 （0.512）

表 6　ProbCut と Null Move Forward Pruning の関係

	NMFP が生じる	NMFP が生じない
ProbCut が生じない	525,612	346,687
α カットが生じる	6,693	0
β カットが生じる	47,786	73,222

表 7　ProbCut と Null Move Forward Pruning の関係（α カットのみ）

	NMFP が生じる	NMFP が生じない
ProbCut が生じない	491,867	485,189
α カットが生じる	22,944	0

5.5 ProbCut と Null Move Forward Pruning の関係

本実験で使用したプログラムは各ノードにおいて Null Move Forward Pruning の判定を行い，枝刈りの判定がされたらそのまま β 値を返す，枝刈りの判定がされなかったら ProbCut の判定に移り枝刈りが発生するか探索が継続するかに分岐する．そこで各ノードで Null Move Forward Pruning が発生するか否かと ProbCut が発生するか否かの関係を調査した．自己対戦と同様の設定で 1 手 3 秒・先手後手ともに変動マージン（2.0σ）で対戦した途中の各ノードの状態を数え上げた．総ノード数が 100 万ノードになるまでの結果を表 6 に示す．また ProbCut を α カットのみにした同様の結果を表 7 に示す．

本実験では ProbCut は一定区間内の深さのみを対象に判定している一方で Null Move Forward Pruning は全ノードを対象に判定を行なっていることから枝刈りの発生数は Null Move Forward Pruning の方が多い．β カットの発生数は多いが，β カットが生じた 121,008 ノード中約 4 割に当たる 47,786 ノードは Null Move Forward Pruning が発生しているノードであることから本来の β カットのみの効力と比較すると効果が薄い．しかし約 6 割は Null Move Forward Pruning で枝刈りできていないノードの枝刈りを行なっており，Null Move Forward Pruning と同時に実装することにはノード数削減の上で十分意味があると言える．一方で α カットは Null Move Forward Pruning が刈ることのできない下側の枝刈りであることから競合はしないが，発生数自体は少ない．

本実験での実装は深さ 5 以上 9 未満を対象とした ProbCut であり，複数の深さで ProbCut が生じる可能性がある．そのため ProbCut を α カットのみにした場合，β カットが起こるだろうノードがカットされず次の一手に進んだとしても手番が変わって α カット側に移る．そのため α カットのみにした表 7 の通り α カットが増える．

6. おわりに

本論文では 3 章において，ゲーム木探索に広く適用されている実現確率探索や静止探索などのゲーム木の縮退・拡張手法が，深浅差の分布推定を困難にしていることを示した．

ProbCut が将棋に有効であることは確認されたが，既存手法からの棋力の有意な向上は見られなかった。しかし固定のマージンを用いた相手には有意性が見られないにしても勝ち越しており，今後適切な特徴やパラメタを用いることで棋力が向上することが期待できる．

今度の課題として，深浅差の推定に有力な特徴の追加が挙げられる．特に SVR を利用しての深浅差の推定のために局面の安定性を十分に説明しうるだけの特徴を見つけることが必要である．

また本論文の自己対戦による棋力の比較は，マージンに関連するパラメタを変えての比較でのみ行なった．枝刈り判定のマージン以外のパラメタや一手の思考時間などその他のパラメタを変動させることで ProbCut が有効である条件の調査を行うことが今後の課題である．

本研究の ProbCut は浅い探索を Null Window 探索により行なったが，3 章で示した通り浅い探索の評価値を通常探索によって得ることで深浅差の分布をより適切に推定できることが予測される．浅い探索に通常探索を行い，その結果を用いて ProbCut の性能を調査する必要がある．

参考文献

[1] Heinz, E.: How DarkThought Plays Chess, *ICCA Journal*, Vol. 20, No. 3, pp. 166–176 (1997).

[2] Buro, M.: ProbCut: An Effective Selective Extension of the alphabeta Algorithm, *ICCA Journal*, Vol. 18, No. 2, pp. 71–76 (1995).

[3] Donninger, C.: Null Move and Deep Search: Selective Search Heuristics for Obtuse Chess Programs, *ICCA Journal*, Vol. 16, No. 3, pp. 137–143 (1993).

[4] Heinz, E.: Extended futility pruning, *ICCA Journal*, Vol. 21, No. 2, pp. 75–83 (1998).

[5] Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R. and Lin, C.-J.: LIBLINEAR: A Library for Large Linear Classification, *Journal of Machine Learning Research*, Vol. 9, pp. 1871–1874 (2008).

[6] 竹内聖悟, 金子知適, 川合慧: 将棋における ProbCut の静止探索への応用, 情報処理学会研究報告, Vol. 2005, No. 87, pp. 9–15 (2005).

[7] Buro, M.: Experiments with Multi-ProbCut and a new high-quality evaluation function for Othello, *Games in AI Research*, pp. 77–96 (1997).

[8] 保木邦仁: コンピュータ将棋の新しい動き : 3. コンピュータ将棋における全幅探索と futility pruning の応用, 情報処理, Vol. 47, No. 8, pp. 884–889 (2006).

[9] 吉原一期, 近山隆: ProbCut の改良と将棋への適用, コンピュータソフトウェア, Vol. 19, No. 3, pp. 201–204 (2002).

[10] 柴原一友, 乾伸雄, 小谷善行: ProbCut の適用有効性 - 局面の分類ごとにパラメータを変えた場合 -, *The 7th Game Programming Workshop (GPW 2002)*, pp. 73–80 (2002).

[11] 伊藤裕, 橋本剛, 橋本隼一: 動的なマージンを用いる Futility Pruning, *The 12th Game Programming Workshow (GPW 2007)*, pp. 1–8 (2007).

[12] Tsuruoka, Y., Yokoyama, D. and Chikayama, T.: Game-Tree Search Algorithm Based on Realization Probability, *ICGA Journal*, Vol. 25, No. 3, pp. 145–152 (2002).