

コンピュータ将棋における SVM を用いた Move Ordering の効率化

野口 拓央^{†1} 古宮 嘉那子^{†2} 並木 美太郎^{†3} 小谷 善行^{†4}

本稿では機械学習手法の 1 種である Support Vector Regression (SVR)を用いたコンピュータ将棋の Move Ordering (MO)を提案する。MO はコンピュータ将棋の探索を効率化のため可能手なるべく良い手から順番に先頭から並べかえる手法である。MO では可能手を 1 手ごとに探索して評価関数を適用すると処理に時間が必要であるため効率化の余地が大きい。そこで本稿では可能手から特徴を抽出して SVR で手の優先度を求めて評価関数を用いずに実行できる MO を考案し、MO を高速化して探索を効率化する手法を提案した。本手法の MO をした後にプロが指した手の MO による平均順位を測定したところ既存手法に 2 手ほど及ばなかったが既存手法と近い結果を出した。また速度が速かったため SVR に評価関数を用いた探索の代替手段となりうる可能性が示唆された。しかし、本手法の性能を確認するために 200 回対局を行ったところ、86-111-3 という結果を得て有意に負け越してしまった。

Efficiency of Move Ordering using SVM in Computer Shogi

TAKUHIRO NOGUCHI,^{†1} KANAKO KOMIYA^{†2} MITARO NAMIKI^{†3} YOSHIYUKI KOTANI^{†4}

This paper proposes Move Ordering (MO) of the computer Shogi using Support Vector Regression (SVR), which is one sort of the watching learning technique.

MO is the technique of arranging moves in the order of value of them to increase efficiency of search of computer Shogi.

There is room for further improvement in efficiency of MO because it is time – consuming; the move search is performed and the evaluation function is applied one by one.

So this paper proposes MO using SVR to speed up MO and enhances efficiency on search. It extracts features of possible moves and gets the priority of moves using SVR.

The average rounds of the professional player's moves were checked to evaluate the method. It showed the MO using SVR can alternate move search and evaluation function in the future because the average \difference of ranks was only two and it is faster.

However, the move search and evaluation function an outperformed MO using SVR significantly when 200 games were performed. The result was 86-111-3.

1. はじめに

近年、コンピュータ将棋は bonanza 法 1) によって飛躍的な進歩を遂げプロ棋士との試合が多く行われるようになった。しかし、未だに人類最強の棋士に勝利できていないためにさらなる進歩が必要だと考えた。現在、コンピュータ将棋では一定の深さまで探索を行って評価関数で評価するという手法が主流であるが、bonanza 法によって評価関数が大きな進歩を遂げたのに対して探索手法は未だに大きな進

歩は存在しない。そこで本研究では現状のコンピュータ将棋の課題は探索面にあると考えた。そして探索面を改善するため探索に評価関数面で有効 1) 2) とされた機械学習を取り入れることを考えた。そこで SVM を可能手の予測に使うことで高速な MO を実現することができないかを考えた。

現在のコンピュータ将棋で使われている MO の既存手法の 1 つに可能手の評価値を 1 手ごとに確認することで評価値の順番に並べ替えるというものが存

^{†1} 東京農工大学大学院 産業技術専攻
Master of Technology Management, Tokyo University of Agriculture and Technology

^{†2} 東京農工大学 工学研究院
Institute of Engineering, Tokyo University of Agriculture and Technology

^{†3} 東京農工大学
Tokyo University of Agriculture and Technology

^{†4} 東京農工大学 工学研究院
Department of Computer and Information Sciences, Tokyo University of Agriculture and Technology

在する．これは1手ごとに評価値を確認する必要があるために低速であることから非効率ではないかと考えた．そこで本研究では機械学習の1種類である SVM を可能手の予測に使うことによって盤面を動かす必要のある評価関数を使用しない MO を提案する．これによって MO の高速化による探索の効率化を目的とした．

2. 関連研究

将棋に機械学習を適用した先行研究の中で最も有名なものに bonanza 学習 1) が挙げられる．この手法は最適制御法に基づき，棋譜の指し手と探索結果が一致するような特徴ベクトルを求める手法である．この手法の開発によってコンピュータ将棋は飛躍的な進歩を遂げるためにコンピュータ将棋の評価関数に機械学習を適用した研究が数多く存在 2) 4) する．またコンピュータ将棋の詰め探索を起動する局面かどうかを機械学習で識別する研究が存在する．3) また機械学習を探索に応用したものとしては FP の安全マージンの値を機械学習によって動的に決定するという研究 5) が存在する．

しかしコンピュータ将棋の MO に機械学習を用いたものは存在しないために本研究にある程度の独自性があると考えた．また，研究 5) では機械学習を適用する際のオーバーヘッドのために勝ち越すことができなかった．そのため本研究では高速化の用途で使用するため効果が期待できるのではないかと考えた．

3. 本研究で使用する手法

3.1 Move Ordering について

本節では本研究で扱う MO について説明する．コンピュータ将棋では一定の範囲を探索してその結果を評価関数で評価するという $\alpha\beta$ 法が主流である．この $\alpha\beta$ 法には最善手を早い段階で発見した場合に探索を効率よく行えるという特徴が存在する．そのため MO は手を良いと考えられる順番に並べ替えることでコンピュータ将棋の探索を効率化する手法の1つである．

図 1, 2 を用いて MO の効果を説明する．丸いノードの中にある数字はそのノードの評価値で，四角いノードの中にある数字は下のノードを探索した結果であるが，まず木の順番が悪く最も後ろに最善手が存在する図 1 では最善手を発見するために全ての局面を探索する必要がある．しかし MO を行い，最善手らしき手が最も前に存在する図 2 の場合では 7 が

最善手であるため，2 番目の局面では 6 を見た時点で α カット，3 番目の局面では 5 を見た時点で α カットできるため 3 つのノードを探索しなくともよい．探索の効率化につながっている．既存手法では，この手の並べ替えのために評価関数を用いているが本研究では可能手ごとに盤面を動かして評価値を算出することが非効率であると考えたために手の評価に機械学習の1種類である SVM を用いることを提案する．

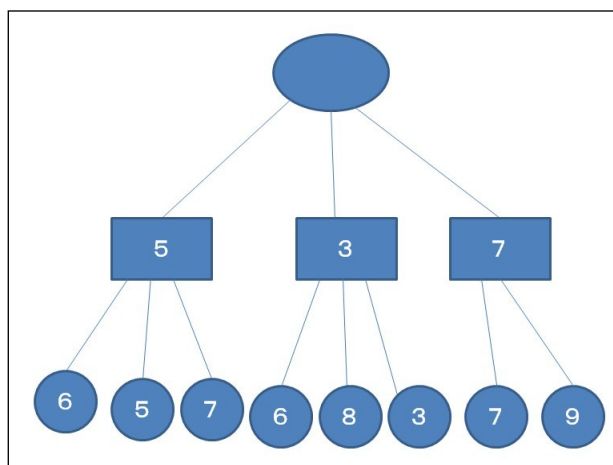


図 1 MO を行っていないゲーム木

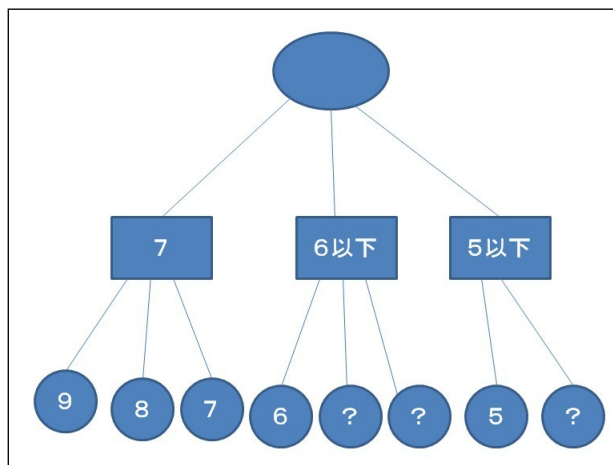


図 2 MO を行っているゲーム木

3.2 SVM について

本手法では可能手を評価する手段として機械学習の1種である SVM を使用することを考えた．SVM は教師あり学習を用いた 2 クラスパターンの識別器をつくる機械学習である．SVM では教師値を使い教師値からの距離が最大となる教師値マージン最大

化超平面を作成することでそれを識別器とする．SVM は汎化能力が高いためにコンピュータ将棋ではよく使われる手法 2)である．しかし 2 値分類ではコンピュータ将棋に応用しにくいいため SVM を回帰分析に使用した SVR がよく用いられている．本手法ではプロの局面からプロの打った手を正例，ランダムに選択したものを負例として回帰分析を行い，出力された関数によって手の有用度を評価することで MO を行う．

4. 提案手法

4.1 SVM を用いた MO について

本論文では MO の高速化するために SVM を用いて可能手を評価する手法を提案する．提案手法は可能手ごとに盤面を動かす必要な評価関数に代わって，SVR による回帰分析を可能手の評価に用いることによる MO の高速化を提案する．本手法では SVM に使う教師値をプロの棋譜から抽出してプロの打った手を正例，そうでないものを負例としてその手がどの程度に打たれるかという指標である優先度を求める関数を回帰分析によって求めた．提案手法はその関数で可能手ごとに優先度を算出して優先度の高い順に並び替えることで MO を行っている．

本手法の既存手法における有意性としてその速度が挙げられる．既存手法は 1 手ずつ可能手ごとに盤面を動かして評価関数で評価しなければならないのに対して，提案手法は盤面を動かすことなく優先度を求める関数で評価するだけでよいため高速に行えることが挙げられる．

本手法を使用するためにはまず機械学習に用いるための特徴を定義する必要があるため以下の特徴を定義した．

本手法の SVM に用いた特徴

- ・成る手であるかどうか(2 値)
- ・味方の王との x 軸距離(0~1, 2, 3~4, 5 より上)
(4 値) (終点のみ)
- ・味方の王との y 軸距離(0~1, 2, 3~4, 5 より上)
(4 値) (終点のみ)
- ・敵の王との x 軸距離(0~1, 2, 3~4, 5 より上)
(4 値) (終点のみ)
- ・敵の王との y 軸距離(0~1, 2, 3~4, 5 より上)
(4 値) (終点のみ)
- ・どんな駒を取ったか
(8 値)

- ・着手によって駒が中央より前から後になったか
(2 値)
- ・着手によって駒が中央より後から前になったか
(2 値)
- ・着手の目的地が，歩が次に成れる位置かどうか
(2 値)
- ・駒の移動先が敵の駒の範囲から抜けたかどうか
(2 値)
- ・駒の移動先が敵の駒の範囲に入ったかどうか
(2 値)
- ・駒の移動先が味方の駒の範囲から抜けたかどうか
(2 値)
- ・駒の移動先が味方の駒の範囲に入ったかどうか
(2 値)
- ・どんな駒の着手をするか
(8 値)
- ・着手の位置が敵陣の 1 マス前かどうか
(2 値)
- ・駒の目的地が一番下かどうか
(2 値)

また本手法では線形カーネル SVM を使用した．これは非線形カーネルと比べると計算が単純である線形カーネルであり高速に優先度を計算できるために MO の高速化を目的とする本研究の目的と一致すると考えたからである．しかし線形カーネルを使用すると駒の種類や進行度など線形で表せない要素を別の手法で表現する必要があるために複数の領域に分割することにした．本手法は可能手を駒種と進行度を基準として具体的には以下のように領域を分割した．

領域を分割する基準

- ・駒種(歩以外の駒 7 種類+突く歩+打つ歩)(9 種類)
- ・進行度(序盤，中盤以降，両方)(3 種類)

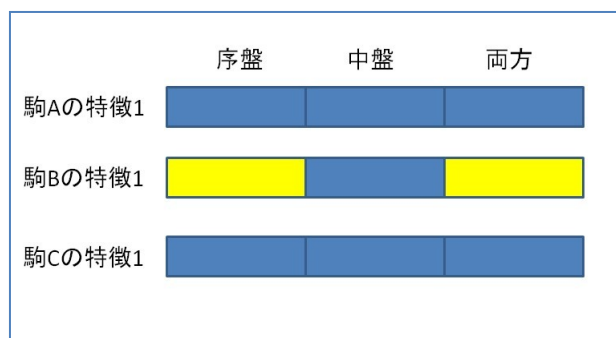


図 3 領域を使用した優先度の計算

そのため本手法では可能手を $9 \times 3 = 27$ 個の領域に分割することになる。そこから優先度を計算する場合は図 3 のように、駒の種類と序盤または中盤以降のうち適合する値と駒の種類と両方の場合の値を合算したものとなる。

領域の分割をこの基準にした理由であるが、駒の種類については、歩に関する可能手が他の駒より多かったために 2 種類に分割した。また進行度についてだが序盤と中盤以降を分ける基準にプログラムについているものを使用した。また序盤と中盤以降に分割した理由であるが終盤の探索は詰め探索が主であり MO の必要性が薄いと考えたからである。

また本手法と既存手法の MO の擬似コードを以下に示す。

既存手法である MO の擬似コード

```
MoveOrdering
(kanoute_list[kanoute_kazu], kanoute_kazu)
{ while ( pi < kanoute_kazu ) {
  Kyokumen_Susumu ( kanoute_list [ i ] );
  // 実際に可能手を指して局面を動かす
  kanoute_list [ i ].hyoka = Hyoka ( )
  // 評価関数で局面を評価する
  Kyokumen_Modosu ( kanoute_list [ i ] );
  // 動かした局面を戻す
}
quick_sort ( kanoute_list [ kanoute_kazu ] )
// 評価値で手を並べ替える
}
```

提案手法である MO の擬似コード

```
MoveOrdering
(kanoute_list [ kanoute_kazu ], kanoute_kazu)
{ while ( pi < kanoute_kazu ) {
  vector = change_vector( kanoute_list [ i ] );
  //可能手を特徴ベクトルに変換
  kanoute_list [ i ].svm
  = predict_value ( kanoute_list [ i ] )
  // 特徴ベクトルを SVR で数値に変換
}
quick_sort ( kanoute_list [ kanoute_kazu ] )
//SVR の数値で手を並べ替える
}
```

まず既存手法の MO は可能手ごとに実際に可能手を指して局面を動かしてから評価関数で局面を評価

しているのに対して、本手法では局面を動かさずに可能手のみで局面を評価している。このため局面を動かす時間と評価関数にかかる時間の差だけ高速な MO が期待できると考えた。

5.実験

5.1 SVM を用いた MO の精度の実験

また MO を行った後の平均順位を用いて既存手法の MO と SVM を用いた MO の精度を比較する実験を行った。まず学習に適切な教師値を調査するために、SVM を用いた MO は正例の教師値をプロの指した手、負例の教師値をランダムに 1 手、2 手、3 手、4 手、全ての指さなかった手、既存手法で MO を行った場合に下位 5 位にくる手と変えて実験を行った。状態を示す特徴は 4. 3 にあるものを使ったがこの時は駒の種類を 8 種類とした。試料にもプロの棋譜を使用し、1500 局のプロの棋譜のうち 1000 局を学習に使い、残り 500 局を試料として使う 3 分割交差検定で行った。さらに付録として新しい特徴を全て入れた後で駒の種類を 9 種類にしたもののデータもここに記録する。既存手法には、評価値を使用した重くて精度が高い MO と、探索結果を使った軽くて精度が低い MO が存在するのだが、探索結果を使った軽い MO は探索という不安定な要素を含んでいて定量的に評価することが難しいため比較対象から外すこととした。すると表 4 のような実験結果になった。

表 4 SVM を用いた MO の精度の実験

教師値	A	B	C
ランダム 1 手	24.11	24.01	23.73
ランダム 2 手	24.61	24.61	24.47
ランダム 3 手	25.34	25.53	25.40
ランダム 4 手	25.94	26.21	25.82
可能手全て	30.49	30.40	30.20
既存手法の 下から 5 位の手	25.11	25.57	23.74
既存手法	18.99	18.91	18.83
特徴改善後	21.19	21.35	21.00
平均可能手総数	82.60	83.87	82.48

(単位は手、小数点第 2 位まで有効)

実験結果に関する考察であるが、既存手法と SVM を用いた MO の精度を比較すると特徴を入れ

る前は 5 手ほどの開きがあることが分かる。これが新しい特徴を入れたものだと差は 2 手くらいの差となる。この数値は可能手総数の平均が 83 手程度なので 5 手の差は決して少ないものでないことが分かるが、評価値を見て評価したものと簡単な特徴から SVR で割り出した数値が 5 手しかないと考えたとこの数値は予想以上に高いとも考え、軽い物との代替に使う場合にはある程度の精度が期待できると考えた。

教師値に関する考察だが、負例をランダムしたもののは数が増えるごとに平均手数が上がる結果となってしまう。これは正例に対する負例の割合が上がった結果、バランスが崩れてしまい学習に悪影響を及ぼした可能性が高い。特に可能性全てを入れたものは 30 手ほどになってしまった。また既存手法の MO の順位が下から 5 位のものを負例の教師値にした場合も負例が 1 つなのにも関わらず、ランダムに 2 手から 3 手くらいの数値をとってしまっている。これは学習する負例の手が偏ってしまうためにランダムよりも悪い結果となってしまうのではないかと考えている。したがって、この結果から SVM を用いた MO は既存手法よりも劣るが十分な精度を持っていると考えられる。そして負例として適切な教師値はランダムに 1 手から選択するのがもっともよいと考えられる。

5.2 SVM を用いた駒ごとの精度の実験

本手法では手の評価に SVR の数値を使用するが、その予備実験として SVM ではどのような駒を動かす手が正確に評価することができるか、できないかの適応性を駒ごとに測定した。測定方法はプロの対局からプロの指した 1 手ごとに、同じ局面のそれとは異なる可能手のうち 2 手を取り出し、それぞれを正例と負例として、特徴に変換して駒ごとに打つ手か打たない手かを 2 分する SVM の識別器を作り、それとは別のプロの対局からプロの指した 1 手ごとに、同じ局面のそれとは異なる可能手のうち 2 手を取り出したものを駒ごとに分別したものを試料としてその手が実際に打たれたか、打たれなかったかの正答率を見ることによって SVM ではどのような駒を動かす手が正確に評価することができるか、できないかの適応性を駒ごとに測定した。

この実験を 1500 局のプロの棋譜を使った 3 分割交差検定によって行った。SVM に使用したライブラリは liblinear とする。すると結果は表 5 になった。

まず、王の正答率の低さについてだが、これは特徴の中に王の強い動きについての特徴がないからだ。これは王がどのような駒を犠牲にしても守るべきものであり、他の駒と扱いが違うものだからである。次に飛車、角の正答率であるが 78% と比較的高いため、SVM の識別が特に苦手とする駒ではないと考える。

次に金だが金は本来の金と、成ったことにより金として扱う駒の 2 種類が存在するため学習が上手くいかないのではないかと懸念していたが思いのほか正答率が高かった。これは金の主な使い方が相手の王を追い詰めることと自分の王を守ることと具体的に言えることと、駒の動かし方が分かりやすい終盤に使われることが理由だと考えている。しかし、銀の場合は王を攻めることの他に中盤に敵陣を攻めることにも使われること、また取られると損害が大きくなりがちな金に対して銀は捨てることが多いために、銀の上手い使い方が特徴に表しにくいために正答率が低くなってしまっていると考えられる。

桂馬も大駒をけん制することや相手の王の行動範囲を狭めるという明確な使い方が存在するため正答率がわりあい高くなっている。香車は中でも動きが単純で使い方が他の駒よりも高いため、正答率が上がっている。最後に歩であるが非常に使い方が多く、局面ごとに重要性が増減し、目的のために捨てる時や逆に絶対に守りぬかないとしない時があるためにこれらの特徴だけでは使い方がわからないために特徴だけでは打つ手か打たない手が判断できないため正答率が低くなってしまっている。

表 5 駒ごとにおける SVM の適応性

駒の種類	SVM A	SVM B	SVM C
王	68.04 %	69.18 %	69.30 %
飛車, 竜王	78.18 %	78.25 %	78.44 %
角行, 竜馬	77.19 %	77.14 %	76.91 %
金の動きをする駒	79.24 %	79.93 %	79.60 %
銀	71.30 %	71.24 %	71.50 %
桂馬	78.62 %	79.13 %	78.22 %
香車	89.97 %	90.36 %	90.22 %
歩	66.35 %	66.43 %	66.03 %

(数値は小数点第 2 位まで算出)

ここから分かることは、SVM で打つ手か打たない手を判別する時に得意な駒は使い方や重要性が安

定している飛車、角行、金、桂馬、香車であり、逆に苦手な駒は他の駒とは全く違う性質を持った王と時と場合によって重要性が変動することが多い銀、歩である。これらの特徴で識別するのは困難であるが、特徴に限らず何らかの形で識別する要素を盛り込むことができればこの識別器の性能を上げることができるのではないかと考えた。そこで歩を動かす手か持ち駒から打つ手かどうかで駒の種類を別のものとして扱うことで安定しない価値を少しでも安定させるべく考えた。また歩は最も下にあるときにある程度の効果を発揮するため、それを特徴に追加した。

また、その結果が正しいかを確認するために平均順位の測定結果から並び替えが上手い/下手な駒と、既存手法で並び替えが上手い/下手な駒の調査を行った。このとき、状態を示す特徴は 4.3 にあるものを使った。MO が上手い/下手な駒の調査を行うために、プロの対局 500 局から 1 局面ごとに可能手を生成して各 MO を行った時に、プロが指した手の順位が既存手法の MO の平均順位に近い値の 19 位を閾値として、閾値を上回った場合の回数を駒ごとに計測した。すると表 6 のようになった。

まず提案手法が平均順位を下回らなかったのは王を動かす手である。これは全体的な精度を見た時に SVM が劣ることを考えると、既存手法では王を動かすことが優先されずに他の手を動かすことが優先されているのに対して、提案手法では王を動かすことが優先されすぎているからではないかと考えた。また飛車と角は提案手法と 100~200 くらいの差しかなかった。これらの駒は SVM を用いた適応性の実験でも 78% と比較的、高い値を出しているために適切な優先度で動いていると考えた。また、金についてはこれも適応性の実験で高い値を出していたために適切な優先度で動いていると考えている。しかし金として扱う成駒は既存手法と比較すると圧倒的に悪かった。これは金と、金として扱う成駒は同じものとして学習しているかだと考える。しかし差が 600 程度なので金として扱う成駒を分けて考えると学習量が不足することと、処理を煩雑になることのデメリットが大きいと考えたためにそのままでもよいと考えた。銀に関しては金よりも適応性の実験で高かったために既存手法よりも SVR の方が閾値を上回る数が少なくなっている。これは桂馬でも同じ現象が起こっている。香車はやや閾値を上回る数が増えたがこれは誤差の範囲内と考えた。しかし、歩ではそれとは反対に適応性の実験で 66% と非常に低い値を出していたために進める手で 1500 手、打つ手では 1800 手くらいの差が存在して

いる。したがってここが既存手法と提案手法にまたがる精度の差となっているのではないかと考え、大きなボトルネックになっているのではないかと考えた。そこで歩に関する手をいかにして打つ手と打たない手に識別するかが本手法を使用する上での課題であると考えた。しかし、この問題を歩の優先度をむやみに増やすことで解決したとしてもその分を他の手が打たれなくなることで精度が下がってしまうおそれがあるため、適切な特徴を考える必要があると考えた。

表 6 プロが指した手の順位が閾値を上回った回数(駒ごと)

平均順位を下回った駒	既存手法 (回数)	SVM を用いた MO (回数)
王	1331	792
飛車	2935	3006
角行	2546	2659
金	2556	1871
金として扱う成駒	268	594
銀	3897	2429
桂馬	1775	1404
香車	772	816
歩 (進める手)	128	1699
歩 (打つ手)	2216	4082
合計	18424	19352

5.4 SVM を用いた MO の速度について

本手法の速度面による有意性を示すために従来手法と本手法の計算時間を測定した。測定手法はプロの棋譜 500 局から 1 局面ごとに可能手を生成して、それらを全て MO し終わった時間を測定した。測定は提案手法、既存手法、何もしない場合で行った。何もしない場合の計算時間は可能手生成時間や棋譜のデータ読み込み時間の参考に測定した。すると表 7 のようになった。

まず、MO 自体の純粋な速度の算出のために測定結果から MO を使用しなかった時の測定結果を差し引くと右のようになった。すると提案手法は既存手法の 2.74 倍の速度で MO ができることが分かる。これは既存手法が可能手 1 手ごとに局面を変化させて評価関数を通して元に戻すという重い処理を行っているのに対して、提案手法は軽い特徴を抽出してそれを事前に学習した結果を足し合わせるだけで処理を行っているために処理が非常に軽いからだと考えられる。したがって本手法は精度面では 4 手ほど劣るものの速度面では圧倒的な優位性があるため、複数回行う必要がある深い位置で行うならばこちらのほうが有用ではないかと考えた。

表 7 500 局の局面から生成される可能手を MO する時間

使用した MO	測定結果	予想される速度
既存手法	6704ms	3532ms
SVM を用いた手法	12844ms	9672ms
MO を使用しない	3172ms	なし

5.5 SVM を用いた MO の対局実験

本手法を使う場合、どのような深さで使えばよいのかを調査するための対局実験を行った。現在、MO は評価関数を用いた MO が残り深さが 3 手以上の場合で行われていて、3 手以下の場所ではそれまで最も良かった手を 2 手目に移動するという簡易な MO を行っているが、提案手法をどの深さで実行すればよいのかを確かめるために既存手法と対局実験を行った。実験に使用したものは深さ 1 手分を 10 分割したものを 1 つの単位とし、既存手法がもともと使われていた深さ 30 (3 手) を基準として、40~30, 35~30, 30~25, 30~20 の場合に提案手法、30 以上で提案手法が使われていない場合に既存手法を使ったものと、30 以上で既存手法のみを使ったものを対局させた場合に提案手法を使う場合も簡易な MO は時間がほとんどかからないため併用した。また、本手法の有意性を確かめるために対局実験を行った。対局は 1 手 5 秒、探索深さは 50 (5 手) で実験には研究室にある対局用ソフトを使用した。さらに本手法の優位性を確認するためにその実験で最も良いとされた深さ 35 以上の場合は既存手法、深さ 25~35 で提案手法を使ったもので行った。これも対局は 1 手 5 秒、探索深さは 5 手 (50) で 200 戦を先手と後手を入れ替えて行った。すると表 8 のような結果になった。

結果から、この手法が最も効果的な深さを考える。はじめに、最も勝率がよかったときに深さ 35~30 の場合である。しかし、この結果は勝ち越したものの有意差はなかった。現在、この手法が速度で勝り、精度で劣っていることを考えると、この深さでは精度で最善手の発見が遅れることによるデメリットと、提案手法を使うメリットが釣り合ったものと考えられる。30~25 の場合も勝ち越してはいないものの引き分けているために、精度が落ちるデメリットと提案手法を使うメリットが釣り合っているものと考えられる。しかし、40~30 になると大幅に負け越しているため、探索深さが浅い場所のときは MO が深い場所での探索に影響を及ぼすために、精度が低くなることのデメリットが大きくなり、負け越してしまったものと考えた。同じように浅い場所で行うようにした 30~20 でも大

幅に負け越していて、MO の回数が増えすぎて、提案手法ですら処理の時間が大きすぎてメリットよりもデメリットが大きくなるのではないかと考えた。したがって、本手法は深さが 35~25 の間で使うことで精度が落ちるデメリットよりも速度が上がるメリットを得ることができ、使うことでメリットが得られる以上の回数、実行されることがなくなるのではないかと考えた。そこで対局実験を行ったところ、82-114-4 で有意に負け越してしまった。

表 8 SVM を用いた MO を使う深さ

提案手法を使用した深さ	戦績	勝率
40~30	83-114-3	42.25%
35~30	99-97-4	52.50%
30~25	98-98-4	50.00%
30~20	68-129-3	34.75%
25~35	82-114-4	42.00%

(勝率は勝利に引分けの半分を加えて、対局数で割ったもの)

6. 考察

本手法は目的とした速度面において MO にかかった時間を見ると、既存手法が 9672ms だったのに対して、本手法は 3532ms で 2.74 倍の速度を出すことができた。これは特徴に先読みした後の結果を使わなかったからと思われる。したがって先読みした後の局面を使うことが MO のボトルネックとなっているためにそれを使わないことができれば既存手法よりも高速な MO を設計することが出来るのではないかと考えられる。

それにも関わらず本手法が負け越してしまった原因であるが、これは、プログラムの評価関数に対して適切な MO が可能な既存手法に対して、評価関数とは全く異なる基準で MO を行う本手法とでは実際に並べ替えをした時に上に来る評価値が違ってしまいうために性能を発揮しきれなかったのではないかと考えている。

このことに関してはまず、提案手法の精度が MO を実行した後のプロが指した手の平均順位を比較したところ既存手法が 19 手程度だったのに対して提案手法の平均順位が 21 程度と 2 手程度の差があった。この 2 手の差が性能に大きな影響を与えていることも考えられる。

これについては提案手法の弱点である歩に関する特徴を強化する必要があるが、歩は使い方が多岐にわた

るため特徴を指定することが難しい。したがって、本手法を実際に使用するときは、評価関数に近い特徴を入れる、教師値をその将棋プログラムの打つ手、打たない手で学習をさせれば実用的になると考えた。また、歩が絡む手の MO が特に弱いため、歩に使い方を上手く分けるための手法を見つけることができれば良い手法になるのではないかと考えた。したがって精度面ではやや劣るものの既存手法と近い結果を出した。

また駒ごとに既存手法の平均順位を下回る駒を調査すると、提案手法は歩に関する手の MO が他と比べて苦手で打つ歩、進める歩を合わせると 3000 手の差が存在して既存手法と大きな差が出るのが分かった。

そこで提案手法を有効に使える深さを調査したところ、深さ 1 手を 10 分割したものを 35~30 の場合が 99-97-4 で、30~25 の場合は 98-98-4 でその他の位置の勝率が悪かったため 35~25 手で既存手法を使った方が良いということが分かった。そこで本手法の有意性を確かめるために実験を行ったところ、82-114-4 という結果だった。これはカイ二乗検定で有意水準 5%において、有意に負け越してしまった。本手法が負け越した理由であるが、これは本手法が行った MO が評価関数を使った既存の MO と違って、使用したコンピュータ将棋プログラムの評価関数とは全く関連性がないために SVM を使った MO の結果と評価関数が乖離していること。また歩に関係する手が精度に悪影響を与えていることも敗因であると考えられる。

したがって今後の課題としては、学習が上手くいきにくい歩を学習するための特徴、またはそれに変わる手法によって精度を上げる方法を考えることがまず必要ではないかと考えた。また、探索木との関係性を深めるために評価関数と関連する要素を学習に取り込むことが必要だと考えた。これに関してはプロの棋譜を教師値にするよりも、提案手法を実装するコンピュータ将棋の棋譜を使った方がよいのかもしれない。これらを解決することができたとき、SVM を用いた MO は有用な手法となると考えられる。

7. おわりに

本論文では探索の効率化という課題のために可能手を 1 手ごとに探索して評価関数を適用することで時間がかかる既存手法の代替として MO の高速化を目的として、なお高精度化が見込める SVM を用いた MO を提案した。まず本手法は従来手法の 2.74 倍の速度を出すことができたため MO の高速化という目的を達成することができた。また結果から提案手法は可能手から簡単な特徴、特に可能手によって変化した後の局面を特徴に入れなくても、既存手法と 2 手しか変わらない精度を出すことができることが判明した。また既存手法で MO した場合は平均 19 手程度の位置にプロが選択する着手があることが分かった。しかし本手法は勝ち越せなかったため探索の効率化という課題を達成できなかった。

参考文献

- 1) 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
- 2) 末廣大輝, 畑埜晃平, 坂内英夫, 瀧本英二, 竹田正幸, SVM によるバイパーティートランキング学習を用いたコンピュータ将棋における評価関数の学習, 電子情報通信学会技術研究報告, IBISML, 情報論的学習理論と機械学習 110 (265), pp. 113-118, 2010
- 3) 三輪誠, 横山大作, 近山隆, SVM による将棋の詰みの予測とその応用, 情報処理学会シンポジウム論文集, pp.143-150, 2004
- 4) 宇賀伸拓也, 小谷善行, 将棋における Tree Strap に基づく評価関数の学習, ゲームプログラミングワークショップ 2010 論文集, pp.114-118, 2010
- 5) 山本一成, 田浦 健次郎, 近山 隆, 学習を用いた枝刈の新手法の提案情報処理学会第 72 回全国大会, 全国大会講演論文集 第 72 回平成 22 年(2), "pp.2-159"- "pp.2-160", 2010
- 6) H. Drucker, C. J. C. Burges, L. Kauffman, A. Smola, and V. Vapnik, "Support vector regression machines," in Neural Inform. Processing Syst. 9, M. C. Mozer, J. I. Joradn, and T. Petsche, Eds. Cambridge, MA: MIT Press, pp. 155-161, 1997,