重ね合わせによるデータ構造を用いた評価関数の学習

築地毅 † 柴原一友 †† 但馬康宏 †† 小谷善行 ††

† 東京農工大学大学院 工学府 情報工学専攻 †† 東京農工大学大学院 工学府

特徴ベクトルに重ね合わせという考え方を導入した新しいデータ構造による学習手法を提案する。Bonanza method の確立により、多くの特徴量を持ち非常に高い精度を持った評価関数の学習に成功している。しかし、既存の方法では学習データにほとんど登場しない特徴の重みは正しく学習されず、結果としてスパースな要素を含む結果となってしまっている可能性がある。そこで将棋において、各特徴と共有している特徴を重ね合わせて学習を行うことでスパースを解決する新しい学習手法を示す。実験の結果、重ね合わせを用いることによって、重ね合わせを用いなかった際に陥るスパースを回避した重みを得ることが出来た。さらに、棋譜との一致率において重ね合わせを用いた学習が高い数値となった。

Learning of Evaluation Function Using a Data Structure from Superposition of Feature Vector

TSUKIJI Tsuyoshi † , SHIBAHARA Kazutomo †† , TAJIMA Yasuhiro †† and KOTANI Yoshiyuki ††

† Department of Computer and Information Sciences,
Graduate school of Technology, Tokyo University of Agriculture and Technology

†† Department of Computer and Information Sciences, Tokyo University of Agriculture and Technology

We propose a learning method by a new data structure which has a superposed feature vector. Recently, an evaluation function which has many features is learned very well by Bonanza method. However, the existing methods have a problem. The weights of the features that hardly appear are not correctly learned and there is a possibility of being a result including sparse elements as a result. The new learning method to solve a sparse data problem by superposing a shared feature to each feature is shown in shogi. The result is that the learning method showed avoiding the sparse elements and raised the concordance rate of the database by professional.

1 はじめに

強いゲームプログラムを作るためには、精度の高い評価関数を設計することが不可欠である。精度の高い評価関数を得るにはより多くの特徴量を持たせる必要があり、多くなるほど手作業による重みベクトルの調節は難しくなるために、機械学習の役割が重要になっている。過去に多くの学習手法が研究されてきたが、2006 年に発表された Bonanza method により、相当数の特徴量を持つ評価関数の学習が可能となった。しかし、既存の手法では学習時にほとんど現れない特徴の重みは正しく学習されず、その結果スパースな要素を含む結果となってしまっている可能性がある。また、多くの特徴量を学習できるようになったため、特徴量をどのように設計するのが良いかを充分に考慮する必要性も出てきた。そこで本稿では将棋において、各特徴と共有した特徴を重ね合わせた新しいデータ構造を考案し、学習時にスパースな要素を回避しやすい評価関数の設計法を提案する。

2 関連研究

ゲームの評価関数の学習において、トップレベルの強さを実現した研究としては、Buro によってオセロの評価関数を最小二乗法によって調節した研究 [3] や、G.Tesauro によってバックギャモンの評価関数を TD 法によって学習した研究 [5] などが挙げられる。TD 法によって学習した例としては他に、Baxter らによるチェスの評価関数の学習 [1] や、Beal らによる将棋の評価関数の学習 [2] などが挙げられる。また、最近では将棋において保木や

金子らによって兄弟局面間による学習[6)9)]が提案され、保木による将棋プログラム Bonanza は 2006 年の世界コンピュータ将棋選手権で優勝という成果を残した。

スパースに対処した研究としては自然言語の分野で、森山らによって枝分かれ同時確率モデルを用いて 2 単語の概念共起確率を、各概念の上位概念を用いて近似する手法[10)]が研究されている。他に Koh らによって、過学習を避けるために l_1- あるいは l_2-regularization を用いた手法[4)]も研究されている。

3　重ね合わせによるデータ構造を用いた学習

3.1　既存の手法の問題点

評価関数の学習手法が確立するに従い、より多くの特徴を持つ評価関数の学習が可能となった。そこで、より強いゲームプログラムを作るためには、特徴量をどのように設計すればよいかが大きな問題となってきた。前述の通り、自己対戦や棋譜などの学習データにほとんど現れない特徴量は学習が正しく進みにくいという問題がある。仮想的に無限に学習データが存在すると仮定すれば、全ての特徴量が無限に存在することになるために正しい重みが得られると考えられるが、現実には有限個の学習データしか得られないために学習データに充分現れない特徴は存在しうると考えられる。ここで本稿では、出現頻度が少ないために、適切な重みを持てない要素のことをスパースな要素と定義することにする。特徴量を単純に増やし学習することで評価関数の精度を高めることを求めると、ほとんど現れない特徴量が更に増えることになるため過学習を起こし、スパースな要素が顕著に表れると予想できる。また、特徴量が多いために学習が進みにくいという問題もあり、学習がより進みやすい設計をする必要もある。このように学習を行うに当たって、特徴量を充分に増やしかつスパースな要素を極力排除した設計をする必要が生じている。また、過学習に対処する手法として l_1- あるいは l_2-regularization という手法がある。l_1- あるいは l_2-regularization は、各重みが発散しないようにそのような重みにペナルティをかける手法である。単に上記の手法で過学習に対処しようとすると、学習データに偏りがあるとほとんど学習データに現れない特徴の重みは 0 となり、学習されるべき価値があるはずの特徴に対しても価値が無いものと学習されてしまう。

3.2　共有特徴と固有特徴

本研究の着想の基点としては、人間がどのように学習をしているかという点であった。そこで、人間が学習をする過程を参考にすることによって、より効率の良い学習手法を考案することを考えた。コンピュータは、細かい特徴を特徴ベクトルとしてたくさん持ち、特徴それぞれに価値を与えた重みベクトルとの線形和によって局面を評価している。すなわち特徴を数多く持ち、かつ特徴それぞれの価値を適切に与えたプログラムが強いプログラムであると言える。対して人間は、特徴を細かく持っている―認識している―わけではないと我々は考えている。例えば、王の位置に対する金の位置の価値は、$81 \times 80 = 6480$ 通りあり、それぞれに対して人間は適切な値を過去の対局などから学習・保持しているとは到底考えられない。実際には人間は「王に対して金の利きがある」「王は自陣にいる」「王と金は 4 マスはなれている」などの、他の状況と共有できる特徴の重みを重ね合わせた結果、王の位置に対する金の位置の価値を得ていると考えるのが妥当である。ここで本稿では、今述べたような他の状況と共有できる特徴のことを共有特徴と呼ぶことにする。また、対照的に他の特徴と共有していない独立した特徴を固有特徴と呼ぶことにする。共有特徴は、充分出現する可能性が高いものでなければ意味がないと考えられる。なぜならば、実際に出現しないような特徴は人間が充分に体験できないため適切な評価が出来ないと考えられるためである。したがって人間は多くの対局や過去の棋譜から、今述べたような違う状況とも共有している共有特徴の適切な重みを適正に学習し、初めて出現する局面に対しても応用できるような評価関数を得ていると考えられる。

3.3　重ね合わせによる評価関数の学習

ここで 3.2 で述べた共有特徴と固有特徴を機械学習に適用することを考える。本稿で提案する重ね合わせによる学習とは、各特徴と共有している特徴それぞれを重ね合わせたデータ構造を用いて

学習する手法のことである。多くの特徴間の関係はそれぞれ排反な関係でなく、何らかの特徴を共有して構成されていると考えられ、本研究ではそのように仮定する。ほとんど登場しない特徴量を他の特徴量と重ね合わせることで、比較的現れる特徴量がほとんど登場しない特徴量を補って学習を進めることができるものである。重ね合わせによる学習では、重みのデータ構造を式 (1) と表して学習を行う。ただし、w_i は特徴 i の重み、$w_i^{'}$ は特徴 i が固有に持つ重み、v_i は特徴 i が他の特徴と共有にもつ重みであるとする。さらに、固有特徴のみの集合を U、固有特徴と共有特徴を含む集合を $U^{'}$ とし、U と $U^{'}$ 間の関係は式 (2) として表されるものとする。また、$S(i)$ を特徴 i が他の特徴と共有にもつ共有特徴集合と定義し、U の要素を一つ入力すると $U^{'}$ の subset を 1 つ返す関数として式 (3) に定義する。

$$w_i = w_i^{'} + \sum_{j \in S(i)} v_j \tag{1}$$

$$U^{'} - U = \{i | i \text{ は共有特徴 } \} \tag{2}$$

$$S : U \rightarrow 2^{U^{'}} \tag{3}$$

実際には、3.2 で述べたように、共有特徴が充分重ねあわされれば固有特徴の重みはほとんど考慮されなくなるのが理想であると考えられる。しかし、どのような組み合わせで共有特徴を重ね合わせれば、充分に固有特徴を表現することができるかがまだ確立できていない。特徴によっては、重ねあわせのみで表現するには向かない特徴も存在する可能性がある。ゆえに本稿では、共有特徴のみで学習することは現段階では非現実的であると考え、固有特徴も学習に用いることにする。

3.4 スパース回避の理論

ここで、重ねあわせによるスパース回避の理論を以下を例にして示す。以下の状況を式 (4) として仮定する。a,b は固有特徴であり、c は共有特徴であるとする。また、a は正しく学習されるのに充分なほど学習データに登場するのに対し、b はほとんど登場しないする。

$$\begin{gathered} U = \{a, b\} \\ U^{'} = \{a, b, c\} \\ c \in S(a) \\ c \in S(b) \end{gathered} \tag{4}$$

重ね合わせの式から、固有特徴 a と b に対する価値 w_a と w_b は式 (5) のように書ける。

$$\begin{aligned} w_a &= w_a^{'} + v_c \\ w_b &= w_b^{'} + v_c \end{aligned} \tag{5}$$

ここで、重ね合わせを行わない場合の評価値について考える。特徴 i の特徴量を x_i とすると式 (6) として書ける。

$$w_a x_a + w_b x_b \tag{6}$$

上記の式 (6) に式 (5) を代入し変形するとと式 (7) を得る。

$$v_c(x_a + x_b) + w_a^{'} x_a + w_b^{'} x_b \tag{7}$$

仮定において、b は a に比べ学習データに現れにくいものとした、すなわち学習がほとんど進まずスパースに陥りやすいものであるとした。すなわち、重ね合わせを行わない式 (6) で学習を行うと x_b がほとんど現れないため、w_b の値が適切に学習されないことになる。一方重ね合わせをした場合、式 (7) より $w_b^{'}$ は x_b がほとんど現れないため学習が進まずペナルティ等により値が 0 となったとしても、x_a が現れやすいために v_c の学習が進むことが分かる。v_c の値が正しく学習されることで、式 (5) より、w_b の値は v_c に重ねあわされた価値が与えられることになり、スパースが回避される。

3.5 重ね合わせの表現方法例

ここでは、重ね合わせの表現方法例について述べる。

3.5.1 駒価値の表現方法例

駒価値を学習する際、成銀などはプロの棋譜を用いて学習すると、頻出する駒ではないためにスパースとなる可能性があると考えられる。そこで、成銀の特徴量と重みベクトルを重ね合わせによって式 (8) の様に表して学習を行うことにする。

$$w_{\text{成銀}} = w_{\text{成銀}}^{'} + v_{\text{金の利き}} \tag{8}$$

式 (8) は、「成銀の価値とは、成銀の固有価値と、金の利きを持つ共有価値の和」と読める。成銀は金と同様の利きを持つために、金の利きを持つ他の駒と共有している特徴を持っていると考え、このように重ね合わせる。

ここで、重ねあわせをしない場合と式の比較を試みる。重ねあわせをしないで、特徴ベクトルと重みベクトルの内積により成銀と金を評価した場合は式 (9) を得る。ただし、x_i を特徴 i の特徴量とする。

$$w_{成銀}x_{成銀} + w_{金}x_{金} \tag{9}$$

式 (9) は、成銀の価値を学習するための特徴量は成銀、金の価値を学習するための特徴量は金、であることを示している。すなわち、学習データに成銀が現れないと成銀の価値は正しく学習されない。対して重ね合わせを用いて評価した場合は式 (10) を得る。

$$\begin{aligned}&(w'_{成銀} + v_{金の利き})x_{成銀} + (w'_{金} + v_{金の利き})x_{金}\\&= v_{金の利き}(x_{成銀} + x_{金}) + w'_{成銀}x_{成銀} + w'_{金}x_{金}\end{aligned} \tag{10}$$

式 (10) は、式 (9) の $w_{成銀}$ に重ね合わせの考え方を導入した式 (8) を代入して変形したものである。式 (10) から、共有特徴である金の利き価値を学習するための特徴量は成銀と金、固有特徴である成銀の価値を学習するための特徴量は成銀、固有特徴である金の価値を学習するための特徴量は金、であることを示している。すなわち、学習データにおいて金が充分に存在すると仮定した場合、金の利きの価値は特徴量である金から正しく学習されることが分かる。実際には、金の利きを学習する時に用いられる特徴量は、成銀と金の他に成桂、成香、と金があるため、学習される機会は相当量増えることになると考えられる。もし、成銀がほとんど登場せず、成銀の固有価値が正しく学習されなかったとしても、金の利きの価値が正しく学習されるため式 (8) から、成銀はスパースな要素になりにくくなると考えられる。

駒価値を学習する時は他に、龍と飛車は飛車の利きを持ち、角と馬は角の利きを持ち、金と成銀と成桂と成香とと金は金の利きを持つ、として学習を行うことにする。

3.5.2 位置価値の表現方法例

ここでは、王とその他の駒の位置関係を学習する際のデータ構造について述べる。王とその他の駒の位置関係は、例えば王と味方の金の位置だけでも $81 \times 80 = 6480$ 通りあるため、学習データに現れずスパースな要素となりやすいと考えられる。そこで、式 (11) のように位置価値を定め、重ねあわせによるデータ構造により学習することを考える。ただし、一般化のため 2 つの駒 p と q がそれぞれ座標 $(x_p, y_p)_p$、$(x_q, y_q)_q$ に存在していると定める。また、$(dx, dy)_{p,q} = (x_q - x_p, y_q - y_p)$ と定義する。

$$w_{(x_p,y_p)_p,(x_q,y_q)_q} = w'_{(x_p,y_p)_p,(x_q,y_q)_q} + v_{(dx,dy)_{p,q}} \tag{11}$$

式 (11) は、「駒 p が $(x_p, y_p)_p$、駒 q が $(x_q, y_q)_q$ に存在する時の価値とは、駒 p が $(x_p, y_p)_p$、駒 q が $(x_q, y_q)_q$ に存在する時の固有価値と、駒 p と駒 q が $(dx, dy)_{p,q}$ だけ離れている時の共有価値の和」と読める。学習データ内において、駒 p が $(x_p, y_p)_p$、駒 q が $(x_q, y_q)_q$ に存在する事例は、組み合わせの数から多くないことは明らかである。そこで、2 つの駒の相対的な位置価値を共有特徴として学習することにする。2 つの駒の相対的な位置価値は、座標ごとに学習させた例と比べて学習データに現れる回数は当然増えるはずなので共有特徴として適切であり、学習は進みやすいと考えられる。また、2 つの駒の相対的な位置価値の情報は、飛車や角の利きがかかっているかといった情報が含まれるため非常に有意であると考えられる。以上より、式 (11) を用いることによって、駒 p が $(x_p, y_p)_p$、駒 q が $(x_q, y_q)_q$ の固有価値が学習されなかったとしても、$(dx, dy)_{p,q}$ の価値が学習されやすいため、少なくとも $(dx, dy)_{p,q}$ の価値を与えられることとなる。なお、この手法に類似した手法が棚瀬によって用いられている [8)]。

3.5.3 自由度の表現方法例

飛車や角、香車などの利きが大きい駒は特に、どれだけ駒が自由に動けるかは局面を評価する際に重要な要素となり得るために、本稿ではそれぞれの駒が移動できる個数を自由度と呼び学習対象とした。しかし、自由度は単純に自由度 1 つ分の価値を自由度で掛け合わせたものではないと考え

られる。例えば、飛車の自由度が全く無い状態である 0 から 1 になった時の価値の上昇量は、自由度が充分ある 15 から 16 になった時の価値の上昇量と比べて大きいと考えられる。そこで、自由度それぞれに対して個別の価値を与える設計にすることにした。ところで自由度は例えば飛車の場合、最小値が 0 最大値が 16 のすべてで 17 通りあり、特に極端に自由度が大きい値は学習データに表れにくいと考えられるため、重ね合わせによる学習を用いてスパースの回避を試みる。本稿では自由度の計算を式 (12) によって、学習を行うことにする。

$$w_{p:\text{自由度}\ i} = w'_{p:\text{自由度}\ i} + \sum_{j=0}^{i-1} v_{p:\text{自由度}\ j} \quad (12)$$

式 (12) は、「駒 p の自由度が i の価値とは、駒 p の自由度が i の時の固有価値と、駒 p の自由度が 0 から i-1 までの共有価値の和と読める。
例えば、自由度 4 を学習する時には、自由度 0 から自由度 3 までの共有価値と自由度 4 の固有価値の和ということになる。すなわち、自由度ごとに特徴を与えそれらの和として学習を行うことで、ほとんど現れないと考えられる自由度が大きい特徴も、その値以下の共有特徴における自由度の和として表現することができる。

この手法は、持ち駒など単純に掛け合わせで表現できない特徴の学習にも応用できると考えられる。

4 実験

重ねあわせによるデータ構造を用いて学習を行った。学習手法は、保木による兄弟局面間における最適制御手法 (Bonanza method)[9] を用いる。ただし、学習時における静止探索は取り合いが発生する時のみ先読みを行うものとし、初期値に駒価値を 500、その他の値に 0 を与えた。また、ラグランジェの未定乗数法を用いずに歩の価値を 200 に固定することにした。

4.1 駒価値の重ね合わせ

王を除く駒の駒価値の学習を 3.5.1 による重ね合わせを用いて行った。学習データとして用いる棋譜はプロの棋譜を 1000 局用意し、プロの棋譜 1000 局を 1 セットとして 15 回学習を行った結果、駒価値は一定の値に落ち着いた。重ねあわせによるデータ構造を用いて学習をさせたものを図 1、比較のため重ねあわせを用いずに学習をさせたものを図 2 に示す。横軸を学習の反復回数、縦軸を価値の値としたものである。なお、参考のため小谷による一般的な駒価値[7] を歩の価値を 200 に正規化して、表 1 に示す。

表 1 一般的な駒価値

飛車	角	金	銀	桂	香	歩
1700	1400	1000	800	600	500	200
龍	馬	成銀	成桂	成香	と金	
2000	1800	1020	1040	1060	1100	

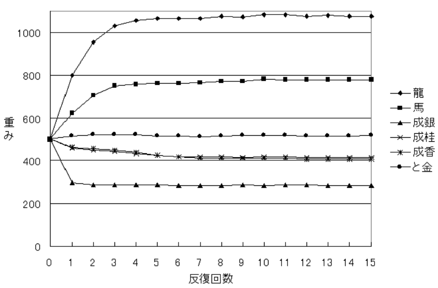

図 1 重ね合わせあり：駒価値

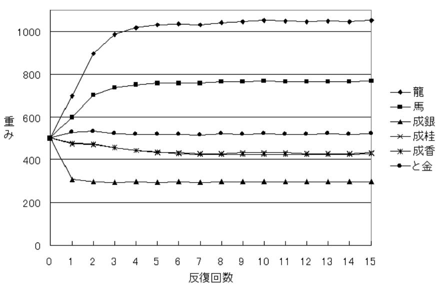

図 2 重ね合わせなし：駒価値

以上より、プロの棋譜を 1000 局用意して学習をさせた場合において、重ねあわせをした場合としなかった場合で、大きな違いは見受けられなかった。これは、駒価値の重みが 13 個に対してプロの棋譜が 1000 局もあり、スパースが起こるほど学習

データが足りないという事態に陥らなかったためと考えられる。以上より、駒価値は、重ね合わせをしても重ね合わせをしなくてもスパース回避には影響がないということが分かり、さらに出現回数が少ないような特徴量を用いて実験する必要があることが分かる。また、今回の結果は表 1 に示した一般的な駒価値に比べて、値が小さいものとなってしまっているため、さらに調節が必要であると考えられる。

4.2　位置価値の重ね合わせ

特徴量を多く持たせた時の学習結果について示す。学習データとして用いる棋譜は、同様にプロの棋譜 1000 局である。学習した特徴は以下の通りである。

- 駒価値
- 持ち駒価値
- 飛車・龍、角・馬、香車自由度
- 王に対する駒の位置価値

本稿では、駒価値と位置価値の重ね合わせによる学習を行うことにし、他の特徴は重ね合わせを行わずに学習を行う。位置価値は、3.5.2 による重ね合わせを用いている。以降簡単のため、駒価値と位置価値の重ね合わせをしたものを「重ね合わせをした学習」、比較のために駒価値と位置価値の重ね合わせをしなかったものを「重ね合わせをしなかった学習」と呼ぶことにする。まず、学習による目的関数の値の推移を図 3 に示す。目的関数の値は保木に倣い、棋譜中の手とその他の可能手による局面の評価値の違いの度合いを表している [9]。横軸をプロの棋譜 1000 局を 1 セットとして学習した反復回数、縦軸を目的関数の値と定める。図 3 から、重ね合わせをした学習と重ね合わせをしなかった学習ともに、目的関数の値が減少していることが分かる。このことは学習が正しく行われた結果、プロの手による局面の評価値が高い値を返し、他の可能手による局面の評価値が低い値を返す傾向を示しており、以上の結果からプロの手に近い評価値が得られていることが示されている。

次に、学習によって得られた駒の位置価値について示す。図 4 と図 5 の自玉の位置は穴熊の位置

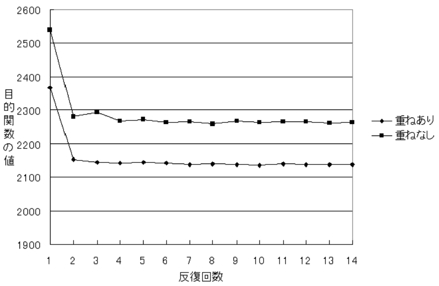

図 3　目的関数の値の推移比較

であり、一般的に 3 九に金があることが多いが実験結果から 3 九の価値が高くなっていることが分かる。また、3 二や 3 四が高くなってしまったのは金で攻めることはあるものの、穴熊における一般的な戦略ではないため、偶然棋譜に特化しただけであると考えられる。

−168	−168	−168	−168	−166	−168	−168	−134	−110
−154	−168	−168	−168	−168	−148	−56	−140	−126
−126	−168	−168	−166	−166	−142	−126	−110	−98
−98	−168	−168	−168	−168	−162	−44	−126	−96
−168	−168	−168	−136	−166	−156	−70	−166	−154
−168	−168	−166	−166	−162	−132	−146	−168	−162
−166	−168	−164	−166	−150	−82	−146	−70	−116
−168	−168	−168	−164	−106	−122	−16	2	−92
−168	−168	−168	−122	−166	−116	−8	−30	王

図 4　重ね合わせあり：自玉が 1 九にいるときの味方の金価値

−84	−84	−84	−84	−82	−84	−84	−52	−26
−70	−84	−84	−84	−84	−66	28	−56	−42
−42	−84	−84	−82	−82	−54	−42	−28	−14
−14	−84	−84	−84	−84	−82	32	−44	−16
−84	−84	−84	−54	−82	−78	0	−82	−70
−84	−84	−82	−82	−80	−54	−66	−84	−84
−84	−84	−80	−82	−76	−40	−80	−42	−84
−84	−84	−84	−80	−30	−50	−10	44	−76
−84	−84	−84	−38	−84	−56	62	−26	王

図 5　重ね合わせなし：自玉が 1 九にいるときの味方の金価値

図 6 と図 7 は、自玉に近いほど王の危険度が高いことを示しており、自玉に近いほど高い値を返している。また、以上の図 4 から図 7 の位置価値は、重ね合わせをした場合としなかった場合とで、値の推移がほぼ一致していることが分かる。例えば図 6 と図 7 の 7 二と 6 三や 3 二と 4 三の価値はいずれも周りの 8 近傍と比較して、高い値を返し

ており、固有価値により値が一定に定められていることが分かる。この結果は、7 二と 6 三は後手の銀冠を、3 二と 4 三は後手の矢倉を学習した結果であると考えられる。なお、今まで挙げた位置価値の例は棋譜上で自玉が存在する場合が多い例であるため、重ね合わせの有無によって大きな違いは見受けられなかった。

-78	-94	-162	-64	-168	-166	-168	-168	-168
-102	-102	6	-58	-46	-152	-66	-168	-168
-154	-92	-60	-6	-86	-20	-168	-164	-168
-138	-130	-84	-108	-70	-108	-162	-168	-168
-166	-146	-130	-120	-142	-132	-162	-168	-168
-154	-52	-158	-130	-162	-162	-166	-168	-140
-104	-34	8	-162	-164	-164	-168	-168	-168
-80	-96	-106	-164	-166	-166	-166	-168	-168
王	-68	-100	-166	-166	-166	-166	-166	-168

図 6　重ね合わせあり：自玉が 9 九にいるときの敵の金価値

-70	-84	-84	2	-84	-84	-84	-84	-84
-84	-82	4	-66	-24	-82	12	-84	-84
-84	-84	-66	6	-58	-8	-84	-80	-84
-84	-80	-66	-66	-44	-64	-84	-84	-84
-82	-84	-84	-82	-76	-68	-82	-84	-84
-70	14	-82	-68	-84	-82	-84	-84	-56
-28	42	84	-84	-84	-84	-84	-84	-84
0	-14	-28	-84	-84	-84	-84	-84	-84
王	14	-18	-84	-84	-84	-84	-84	-84

図 7　重ね合わせなし：自玉が 9 九にいるときの敵の金価値

一方、図 8 と図 9 は、自玉が 6 一にあるというプロの棋譜において非常に稀な例である。今回用いた学習データにはほとんど登場しなかったため、重ね合わせを用いなかった図 9 の価値はすべて 0 となってしまっている。一方で、重ね合わせによって学習した図 8 は、共有価値の分だけ値を得ることに成功している。値が高いところが、先手にとって危険な位置を表しているため、自玉に近いほど自玉にとって危険、すなわち値が高い傾向を見ることができる。また、7 二の価値が回りより高くなっており自玉が 7 一や 6 二に移動することを防ぐ手の評価値が高くなっている。今述べた、王の移動可能位置を減らすような価値は、王が 6 一にある時固有の価値ではなく、どの局面においても共有できる価値であると考えられる。また 9 段目における値が高いのは、共有特徴であっても自玉が一段目にいて敵飛車が九段目にいることが稀であり、学習が適切に進んでいないためと考えられる。

また、本稿における位置価値はマイナス方向に学習が進んでいるものが多かったように思われる。これは、Bonanza method の特性においてプロで指されなかった手がすべて不正解であるとされてしまうため、マイナス方向に偏りやすくなっていてしまったと考えられる。特に、重ね合わせを用いることで不正解によるマイナスが重ね合わせれてしまい、値がマイナス化することが予想され、実際にそのような結果となってしまっている。適切なペナルティをかけることで、値の変動幅を調節するなどの微調整が課題として挙げられる。

-82	-78	-82	王	-84	-80	-80	-82	-82
-84	-82	-62	-78	-76	-82	-80	-82	-82
-84	-84	-82	-84	-84	-84	-80	-84	-84
-84	-84	-82	-84	-84	-84	-82	-84	-84
-84	-84	-82	-82	-82	-84	-84	-82	-84
-42	-64	-74	-82	-82	-84	-84	-84	-84
-56	-56	-58	-56	-82	-84	-84	-84	-84
-56	-42	-42	-56	-82	-84	-84	-84	-70
-14	-14	-14	-14	-56	-56	-56	-28	-42

図 8　重ね合わせあり：自玉が 6 一にいるときの敵の飛車価値

0	0	0	王	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

図 9　重ね合わせなし：自玉が 6 一にいるときの敵の飛車価値

最後に、プロの棋譜との一致率を図 10 として示す。ただし、一致率計算に用いるテスト用データは学習用データと同一のものを用い、学習と並列に一致率を計算している。図 10 から、重ね合わせをした学習と重ね合わせをしなかった学習ともに、学習が進むについて一致率が向上している傾向が見られる。また、一致率は重ね合わせををした学習の方が高くなり、より精度が高い評価関数を得られる結果となった。特徴量をスパースが発生するほど多量に設計し学習を行うと、重ね合わせをした学習の方が精度が高い評価関数を得られるこ

とが分かった。4.1 で、スパースな要素がほとんど現れない特徴では重ね合わせによる利点は示せなかったが、スパースな要素が現れやすいと考えられる位置価値の学習においては重ね合わせによる利点を示すことができたことから、スパースが顕著になる特徴においてヒューリスティックに適切な重ね合わせを設計することにより、より精度が高い評価関数を得られることを示すことができた。

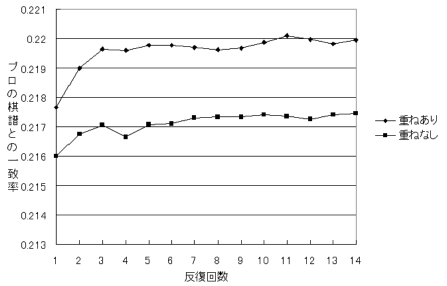

図 10　プロの棋譜との一致率

5　おわりに

本稿では、重ね合わせという新しいデータ構造による学習手法を提案した。重ね合わせを用いることで、スパースな要素に対して共有できる価値を重ね合わせることでスパースを回避し、特徴が多い環境において精度の高い評価関数を得ることに成功した。今後の課題としては、今回の実験では特徴が他のプログラムと比べ貧弱なものとなっていたため、特徴をより増やした時に重ね合わせを行うことで、精度や収束の速さにどれだけの違いが見られるかを、学習実験や対局実験により示す必要がある。また、共有特徴をどのように用意して重ね合わせるかはヒューリスティックを用いているため、どのように重ね合わせることでより精度が良い評価関数を得られるかについて実験をする必要もある。さらに、共有特徴の重ね合わせにより極端にマイナス化するなどの傾向を回避する手法を考え出さなければならない。

参考文献

1) J.Baxter, A.Tridgell, L.Weaver : Experiments in Parameter Learning using Temporal Differences, ICCA Journal, Vol.21, No.2, pp.84–99, 1998

2) D.F.Beal, M.C.Smith : First Results from Using Temporal Difference Learning in Shogi, LNCS 1558, pp.113–125, 1999

3) M.Buro : Improving heuristic mini-max search by supervised learning, Artificial Intelligence, Vol.134, No.1.2, pp.85–99, 2002

4) K.Koh, S.-J. Kim, S.Boyd : A method for largescale l1-regularized logistic regression, In 22nd National Conference on Artificial Intelligence, pp.565–571, 2007

5) G.Tesauro : Temporal Difference Learning and TD-Gammon, Communications of the ACM, Vol.38, pp.58–68, 1995

6) 金子知適，田中哲朗，山口和紀，川合慧：駒の関係を利用した将棋の評価関数の学習, 情報処理学会論文誌, Vol.48 No.11, pp.3438–3445, 2007

7) 小谷善行：コンピュータ将棋の頭脳, サイエンス社, 2007

8) 棚瀬寧 ： 棚瀬将棋の技術背景, 情報処理, vol.49, No.8 Aug, pp987–992, 2008

9) 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, pp.78–83, 2006

10) 森山健太, 但馬康宏, 藤本浩司, 小谷善行：枝分かれ同時確率モデルを用いた「A の B」の意味分類, 情報処理学会 自然言語処理研究会研究報告, 2007-NL-183, pp101–106, 2008