

木カーネルを用いたSVMによる麻雀打ち手の順位学習

三木 理斗^{†1} 三輪 誠^{†2} 近山 隆^{†1}

ゲームの評価関数における特徴要素の作成は、そのゲームについての深い知識を必要とするため難しい課題である。そこであまりゲームに依存しない簡単な特徴構造をもとにして高精度な評価関数を作成する汎用的な手法が求められる。本研究ではカーネル法を用いたSVMによって麻雀のツモ局面における打ち手の順位づけの学習を行った。麻雀の手牌の構成を木構造の一種と見なし、木カーネルを用いて手牌間の類似度を計算することによって打ち手の分類を学習した。熟練者の牌譜を用いて実験したところ、牌譜の打ち手と予測された最善手の一致率は51%となった。

Learning to rank moves in mahjong using SVM with tree kernels

AYATO MIKI,^{†1} MAKOTO MIWA^{†2} and TAKASHI CHIKAYAMA ^{†1}

It is very difficult to create evaluation features in computer game players because it requires expert knowledge for the game. So we have to find a new way to generate high-performance evaluation functions from simple structural features. In this paper, we present the method of learning to rank moves in mahjong using SVM with kernel method. We extracted a tree structure from a hand of mahjong and classify moves by calculating a similarity of hands with tree kernel. As a result of the experiment using expert game records, the classifier predicted the expert moves with 51% accuracy.

1. はじめに

コンピュータゲームプレイヤーにおいて、局面の有利さを見積もる評価関数は非常に重要な要素である。評価関数を人手で作成するにはゲームについての深い知識と多大な労力を必要とするため、近年では機械学習の手法を応用して、コンピュータに評価関数を自動的に習得させる研究が多く行われている。これまでのゲームにおける評価関数の学習手法には、特徴要素を作ってその重みづけを学習するものが多い。しかし特徴要素の作成もまたゲームの深い知識を必要とし、盤面の特徴を過不足なく表現できるような特徴要素を作成することは難しい。またそのような特徴要素はゲームの性質に大きく依存し、統一的な扱いがしづらいという問題もある。

この問題を解決する方法の一つとして、分類器の一種である Support Vector Machine¹⁾(以下 SVM)などで用いられており、多くの問題で有効であるとして

注目されているカーネル法を利用することが考えられる。カーネル法とは、通常はデータを特徴空間に写像して特徴空間上での内積(類似度)を用いて分類を行わなければならないところを、内積に置き換わるような値を元のデータから直に求めるカーネル関数を使うことで、写像を陽に行わずに特徴空間上での線形分類を行う手法である。これをゲームに応用することで、もし局面の特徴的な類似度を表すようなカーネル関数を利用することができれば、この類似度に対応する特徴要素を明示的に生成せずに局面から直接評価値を求める評価関数を学習することができると考えられる。

本稿では麻雀における手牌の評価関数の学習を目的として研究を行った。方法としては、手牌の構成を一種の木構造と見なし、カーネルの一種である木カーネルを用いた非線形 SVM によって手牌の分類を学習した。訓練データとして熟練者の牌譜を用い、牌譜で打たれた手とそれ以外の手を分類するような分類器を学習することで、熟練者に近い手を持つような評価関数を作成できると考えられる。

木カーネルによる手牌の分類を評価関数に利用すると評価関数の処理時間が非常に大きくなると予想されるが、麻雀は将棋や囲碁のような深い探索を必要としないゲームなので、評価関数にかかるコストの大きさ

^{†1} 東京大学大学院工学系研究科

Graduate School of Engineering, The University of Tokyo

^{†2} 東京大学大学院情報理工学系研究科

Graduate School of Information Science and Technology, The University of Tokyo

としては十分に許容できる範囲であると考えられる。

実験において、ツモ局面における熟練者の打ち手と分類器の予測した最善手との一致率は51%となり、人手の特徴要素(自分の手牌以外の情報を含む)を用いてニューラルネットワークで学習した場合²⁾の一致率56%にかなり近づく結果となった。

以降、2章で関連研究について紹介し、3章で本研究の手法を示す。そして4章で実験結果について示し、最後に5章でまとめと今後の課題を述べる。

2. 関連研究

2.1 単純な特徴からの機械学習

人手で高度な特徴要素を設計せずに機械学習によって高い性能を獲得したコンピュータゲームプレイヤーの研究例としては、Tesauro³⁾によるバックギャモンのプレイヤーであるTD-Gammonがある。TD-GammonでTesauroは強化学習⁴⁾の一種であるTD学習によって評価関数のパラメータを学習した。TD-Gammonの初期の設計で使用された特徴要素は単純に盤上の駒の配置を3値で表現したものであり、高度な戦略的解釈を用いた特徴要素は一切含まれていなかった。しかし、約30万試合の自己対戦の後、TD-Gammonはそれまでの最強のコンピュータプレイヤーであったNeurogammonとほぼ互角の性能を獲得した。

2.2 麻雀に関する研究

北川ら²⁾は最適制御理論に基づいて牌譜の打ち手と評価関数による最善手の一致度の誤差を最小化することを目指し、3層ニューラルネットワークを用いて打ち手の評価関数を学習する研究を行った。ニューラルネットワークの入力層には人手で作成した1,532のbooleanの特徴要素(表1)を使用し、6,079試合分の牌譜からツモ局面については575,906局面を学習データとして用い、学習の繰り返し数を40回まで増やしながら実験を行っている。結果は図1のようになり、最大で56%の一致率に到達した。

3. 木カーネルを用いた打ち手の順位学習

この章では本研究の具体的な手法について述べる。最初に麻雀の手牌を木構造に変換する方法について述べ、次に木カーネルを用いた順位の学習のアルゴリズムと実装について説明する。

3.1 手牌の木構造の抽出

麻雀の手牌は構成している要素を分解していくことによって木構造で表現できると考えられる。図2の手牌を例にして、木構造の具体的な抽出法を図3に示す。最初に根ノードとして「手牌」があり、その下に

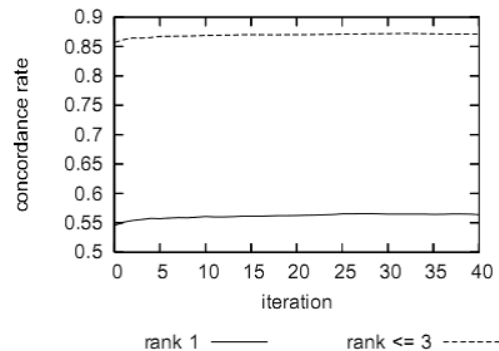


図1 ツモ局面の一致率 [北川, 2007]

自分の状態	面前の持ち牌
	面前の持ち牌 2 枚の組み合わせ
	面前の持ち牌 3 枚の組み合わせ
	鳴いた牌の構成と状態
	面子数
	両面搭子数
	カンチャン搭子とペンチャン搭子の和
	対子数
	テンパイしているかどうか
	ドラの枚数
	面前であるかどうか
	親であるかどうか
	リーチしているかどうか
自分が捨てたことのある牌	
他プレイヤーの状態	鳴いた牌の構成と状態
	鳴いた回数
	鳴いた牌の中で見えているドラの枚数
	親であるかどうか
	リーチしているかどうか
	そのプレイヤーに対する完全安牌
場の状態	筋や壁などによって安全度が高い牌
	自分との点差
	オーラスかどうか
	見えていない牌の残り枚数

表1 麻雀の評価要素 [北川, 2007]

「面子」、「面子候補」、「孤立牌」の3つのノードがある。さらに「面子」の下には「暗刻」、「明槓」などのノードが付き、「面子候補」の下には「対子」、「両面塔子」などのノードが付く。そしてその下に「字牌の暗刻」、「萬子の両面塔子」などの種類ごとに分かれたノードが付き、最後に葉ノードとして具体的な「東の暗刻」、「三四萬の両面塔子」、「六索の孤立牌」などが付く。



図 2 手牌の例

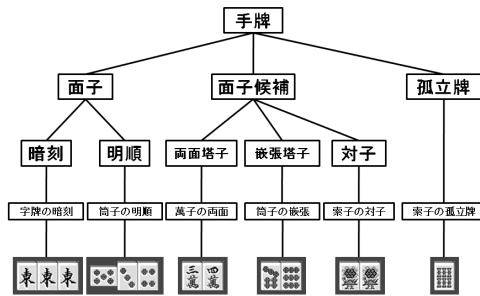


図 3 手牌の木構造の抽出

ポンやチーによる鳴き牌のノードは固定されるが、それ以外の純手牌は解釈によって複数の木構造を取りうる。例えば三萬が二つと四萬が一つあった場合、「三萬の対子」と「四萬の孤立牌」の組み合わせと見ることもできるし、「三四萬の両面塔子」と「三萬の孤立牌」の組み合わせと見ることもできる。それらをすべて考慮していたのでは組み合わせが膨大になってしまうので、本研究では向聴数（あがりまでの必要牌の数）が最も小さく、さらに面子および面子候補について適宜決めた順序が最も若い一つのみを抽出した。

3.2 木カーネルによる順位の学習

木カーネルによる木構造の順位の学習は、Moschittiらによる研究⁵⁾の手法に従った。Moschittiらは自然言語の構文木について、ある述語とそれに対応する複数の項との結び付きを木カーネルを用いて順位付けする研究を行っており、そこで用いられている手法を利用した。

まず木カーネルの計算は次のようにして行う。ある木の部分木空間 \mathcal{F} が与えられたとして、

$$\mathcal{F} = \{f_1, f_2, \dots, f_{|\mathcal{F}|}\} \quad (1)$$

f_i がノード n を祖先に持つときに 1、それ以外るときに 0 をとる関数 $I_i(n)$ を定義する。その上で二つの木 t_1 と t_2 のカーネル関数は (2) 式ようになる。

$$K_t(t_1, t_2) = \sum_{n_1 \in N_{t_1}} \sum_{n_2 \in N_{t_2}} \Delta(n_1, n_2) \quad (2)$$

N_{t_1} と N_{t_2} はそれぞれ木 t_1 と t_2 のノード集合で、関数 Δ は (3) 式で定義される。

$$\Delta(n_1, n_2) = \sum_{i=1}^{|\mathcal{F}|} \lambda^{l(f_i)} I_i(n_1) I_i(n_2) \quad (3)$$

$0 \leq \lambda \leq 1$ で、 $l(f_i)$ は部分木 f_i の深さである。 $\lambda^{l(f_i)}$ は大きな部分木の影響を制御するパラメータで、 $\lambda = 1$ のとき Δ は単純に n_1 と n_2 を根とする二つの木の共通部分木の数を表す。なお、 Δ は $O(|N_{t_1}| \times |N_{t_2}|)$ で計算可能である⁶⁾ ため、木カーネル関数を用いる手法は比較的高速であると言える。

次に二つの木をペアとして訓練サンプルを作る。

$$e_i = \langle t_i^1, t_i^2 \rangle \quad (4)$$

このサンプルペアは第 1 項の方がスコアが高ければ +1、第 2 項の方がスコアが高ければ -1 とラベル付けされる。そしてサンプルペア同士のカーネル関数を次のように定義する。

$$K_{tr}(e_1, e_2) = K_t(t_1^1, t_2^1) + K_t(t_1^2, t_2^2) - K_t(t_1^1, t_2^2) - K_t(t_1^2, t_2^1) \quad (5)$$

このカーネル関数はペアに含まれる二つの木の順位関係の類似度を表し、これを SVM などを利用してすることによって「第 1 項の方が大きいクラス」と「第 2 項の方が大きいクラス」というようにペアを分類する分類器を学習することができる。

3.3 カーネル法を用いた SVM

ここでは SVM と、SVM で非線形分離問題を解くときに用いられるカーネル法について説明する。

3.3.1 SVM

SVM¹⁾ は、1992 年に Vapnik らによって提案された 2 クラスの分類器の 1 つである。SVM は、超平面とそれに最も近い例との距離（マージン）を最大化するように、訓練データを線形分離する（分離）超平面を求める。

分離超平面の法線ベクトルを w とし、識別関数を $g(x) = w \cdot x + b$ とすると図 4 のようになる。 y_i を x_i が正例なら 1、負例なら -1 をとる数とすると、 $y_i(w \cdot x + b) \geq 1$ の条件の下でマージン $2/\|w\|$ を最大化すればよいということになる。ラグランジュの未定乗数法を用いることでマージン最大化は $\sum_{i=1}^n \alpha_i y_i = 0$

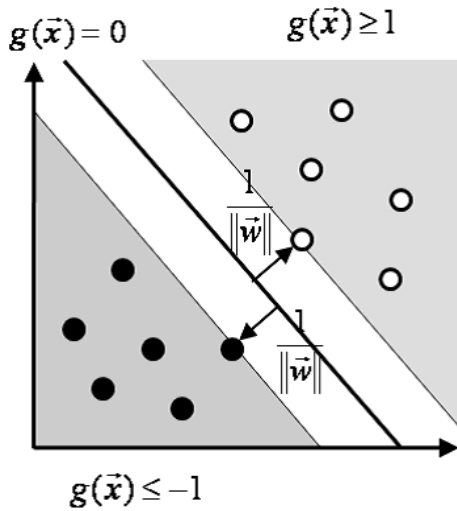


図 4 Support Vector Machine

および $\forall_i \alpha_i \geq 0$ の条件下で (6) 式を最大化する問題に置き換えられる。

$$W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \quad (6)$$

3.3.2 カーネル法

問題が線形分離可能でないときには一度データを高次の特徴空間に非線形写像で移してから、その特徴空間上で線形分離を行う方法がある。データ \mathbf{x} の特徴空間への非線形写像を $\phi(\mathbf{x})$ とすると、(6) 式は (7) 式のようにになる。

$$W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) \quad (7)$$

ここで内積 $\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$ に置き換わるようなカーネル関数 $K(\mathbf{x}_i, \mathbf{x}_j)$ を代わりに用いることで、特徴空間への非線形写像を陽に行わずに特徴空間での分類を行う手法をカーネル法と呼ぶ。置き換えが可能なカーネル関数としては、木カーネル以外に以下のようなものが知られている。

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + l)^p \quad (8)$$

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}\right) \quad (9)$$

(8) 式は多項式カーネル、(9) 式は RBF カーネル (ガウスカーネル) と呼ばれる。

3.4 学習データ

学習データの作成について述べる。牌譜の中であるツモの直後の局面に注目する。その局面で、牌譜の手を打って残った 13 枚の手牌の木 T_0 と、牌譜で打たれなかった手を打って残った 13 枚の手牌の木 $T_1 \dots T_n$ を生成する。次に二つの木からなるペアを作る。ペアは第 1 項の方が順位が高いものを正例として、 T_0 を第 1 項とした n 個のペア $(T_0, T_1), \dots, (T_0, T_n)$ を作り、また同様に第 2 項の方が順位が高いものを負例として、 T_0 を第 2 項とした n 個のペア $(T_1, T_0), \dots, (T_n, T_0)$ を作った。

木の生成は牌譜の各局面について行うが、次のような注意点がある。麻雀では攻めている時と降りている時で最適な評価関数が全く異なると考えられる。したがって学習でもそれを考慮する必要があるが、現在攻めているのか降りているのかを大量の牌譜について判定するのは非常に難しい。そこで本研究では、誰もリーチまたは三鳴きしていない局面は全員が攻めているものと見なし、そのような局面のみを使って攻めの評価関数のみを学習することとした。

3.5 順位の判定

手の順位の判定は以下のようにして行う。テストデータとして、全ての手の木をペアにして分類器に与える。ある手 i の木 T_i を含むペア

$(T_i, T_0), \dots, (T_i, T_{i-1}), (T_i, T_{i+1}), \dots, (T_i, T_n)$ の分類結果のうち、負の値であったものの個数+1 がその手 i の予測された順位に対応する。

4. 実験

4.1 実験方法

4.1.1 実験環境

実験は CPU Dual-Core AMD Opteron 2.4GHz、メモリ 32GB のマシン環境で行った。

4.1.2 実験用の実装

学習用の SVM の実装としては、Moschitti によって作成された SVM-Light に木カーネルを実装した改良版⁷⁾(以下 SVM-Light-TK) を用いた。SVM-Light-TK では 3.2 節の方法に従って、二つの木のペアを与えるとどちらの木が上位であるかのランキングを判定するような分類器を学習することができる⁵⁾⁸⁾。これを用いて学習データの正例を 1、負例を -1 として学習を行った。学習された分類器は、同様に木のペアを与えると、第 1 項の木が上位の場合は正の値を返し、第 2 項が上位の場合は負の値を返す。したがって、3.5 節で説明したようにある局面で生成される木のペアをすべて与え、上位と判定された数が多い順に並べれば手

をランキング付けすることができる。

SVMのコストパラメータ C は 0.1、木カーネルの減衰パラメータ λ は 0.4 に設定した。牌譜から訓練及びテスト用の木構造データを作成する部分、分類結果から順位を判定する部分は Python で実装した。

4.1.3 実験用牌譜

実験用データとして、システムティック麻雀研究所⁹⁾で公開されている東風荘の牌譜のうち、とつげき東北氏のツモ局面のみを用いた。この牌譜から、牌譜の打ち手とそれ以外の打ち手の木構造データを抽出した。得られた木の数は 100 試合分の場合で 60,565 であった。訓練データは牌譜の打ち手の木に対してそれ以外の手の木ひとつひとつのペアを作成し、テストデータはランキングの判定のために全ての手について全体でペアを作成した。

4.1.4 学習の評価

学習の評価は、牌譜の打ち手と学習された分類器の予測した最善手との一致率について、4-fold cross validation で行った。

4.2 実験結果

図 5 に牌譜の打ち手と分類器によって予測された最善手の一致率の遷移を示す。横軸は学習に用いた牌譜の局面数である。凡例の $rank = 1$ は分類器によって予測された牌譜の打ち手の順位が 1 位だった割合、すなわち一致率である。 $rank \geq 2$ は分類器によって予測された牌譜の打ち手の順位が 2 位以内だった割合 (2 位以内率)、 $rank \geq 3$ 、 $rank \geq 5$ は同じく 3 位以内だった割合 (3 位以内率)、5 位以内だった割合 (5 位以内率) である。学習局面数 4,182(100 試合分) で、一致率は約 51%、3 位以内率は約 85%、5 位以内率は約 94% となった。一致率はまだ上昇を続けているので学習局面数はまだ不足していると考えられるが、これ以上の局面数での学習データはメモリ不足によって未実験のため取れていない。

4.3 予測に失敗した局面

最善手の予測に失敗した局面について具体例を挙げて詳しく見てみる。図 6,7,8 は牌譜の打ち手の予測順位が 10 位以下という低い順位だったケースである。

図 6 の局面については、牌譜ではチャンタという役を狙う意図を含めて七筒を切ったと考えられるが、学習では七筒の対子を面子候補の一つとして保持しようとし、また一二萬はペンチャンとして木構造が作られているため、残った一萬を不要な孤立牌として切るように判断してしまっている。このケースでは「役の知識がない」、「一一二萬をまとまった形として認識できない」という二つの問題点が浮き彫りとなった。

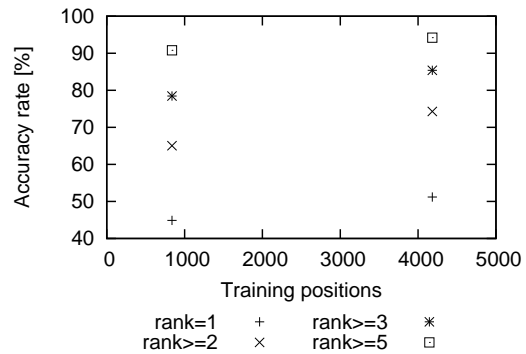


図 5 木カーネルによる一致率

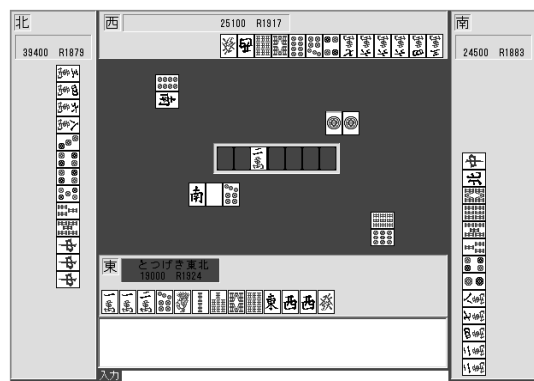


図 6 予測 10 位以下の局面 1 予測 1 位は「一萬」

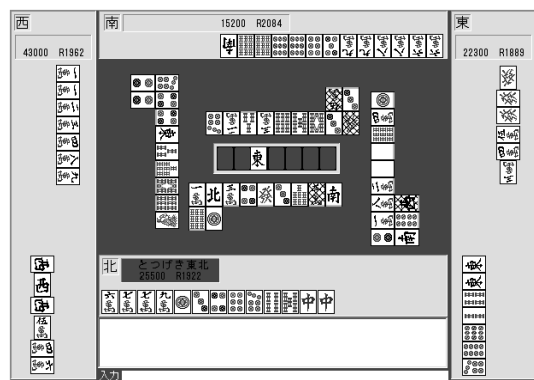


図 7 予測 10 位以下の局面 2 予測 1 位は「九萬」

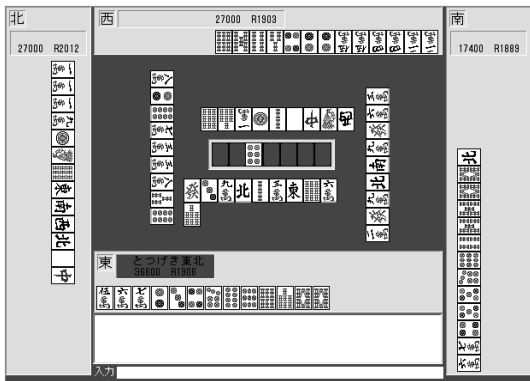


図 8 予測 10 位以下の局面 3 予測 1 位は「六索」

図 7 の局面については、かなり順目が進んでいるがあがり近くに、上家と下家の攻撃の様子が強いことから降りていると考えられる。牌譜では安全な一筒を切ったが、学習では普通の攻めの場合と同様の判断で九萬が最善手となった。上家が萬子の一色手を狙っているのは非常に危険な判断である。このケースでは「敵の鳴き・捨て牌の情報がない」、「降りる状況が判断できない」という制約が失敗の原因となった例であると考えられる。

図 8 の局面については、木構造の作り方における問題が表面化している。牌譜では七索を切ることで六七策の両面待ちにしているが、学習では六索を切ってシャボ待ち(対子二つの待ち)にしてしまっている。これは木構造にしたときに牌譜の手では「六七八索の順子」と「八索の孤立牌」として扱われるため単騎待ちに見えてしまい、シャボ待ちより評価が低くなってしまったと考えられる。木を作る時に対子よりも順子の優先順位が高くなっているために起こる認識ミスであるが、このように両面待ちのケースを見逃すことのないように、より適切に手牌の構造を解釈する必要があると考えられる。

ちなみに、牌譜の打ち手の予測順位が 2 位と惜しい結果になったケースは、二つの不要な孤立牌がありどちらを切るか、という局面がほとんどであった。

4.4 他手法との比較

4.4.1 北川らの研究との比較

人手の特徴要素を用いて 3 層ニューラルネットワークで学習した北川らの研究ではツモ局面における手の一致率は 56%であったのに対し、本研究では 51%と 5%ほど低い結果となった。この原因としては、まず学習局面数が少ないことが考えられる。北川らの研究では 6,079 試合分の牌譜から 575,906 局面を用いたのに対し、本研究では 100 試合分、4,182 局面を用いる

にとどまった。木カーネルを用いた SVM はニューラルネットワークや通常の線形 SVM に対して非常にメモリ消費量が多いため、多くの局面を用いて学習を行うことが難しいという問題がある。

他の原因としては、相手の捨て牌やドラ、親番などの自分の手牌以外の状況を全く考慮していないことが挙げられる。しかし、逆に手牌の情報のみからここまでの精度を得ることができたとも言え、さらに他の情報を組み入れることでより精度が上がるのが期待できる。木カーネルと線形の特徴要素を組み合わせたり、木カーネルと他のカーネルを組み合わせたりすることは比較的容易に可能なので、そのような手法を試すことは有用であると考えられる。

また、4.3 節で示したように抽出した木構造の精度があまり高くなかったことも原因として考えられる。本研究では麻雀の手牌に対して一つの木構造しか対応させていないが、実際に人間はもっと柔軟な解釈を行っている。重要な牌の組み合わせを見逃さないように、より洗練された構造を検討する必要があると考えられる。

4.4.2 特徴要素を用いた線形 SVM による学習との比較

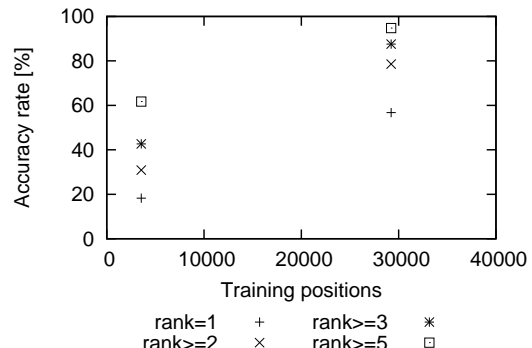


図 9 線形 SVM による一致率

北川らの研究と同じ特徴要素を用いて線形 SVM で学習した場合の結果を図 9 に示す。学習データには同じ牌譜から最大で 285 試合分、29,232 局面を用いた。実装として SVM-Light¹⁰⁾ の Ranking SVM¹¹⁾ を使用した。コストパラメータ C は 0.1、最適化の閾値 ϵ は 0.1 に設定した。結果として一致率は最高で約 57%となった。こちらはまだ一致率は飽和しておらず、さらに大きな局面数での実験が必要であると考えられる。

以上により、学習局面数がほぼ同じ場合には木カーネルの方が高い精度を示すことがわかった。また、よ

り少ない局面数で3層ニューラルネットワークを用いた場合とほぼ同じ性能に到達できることがわかった。

しかし木カーネルで4,182局面を学習する場合と線形で29,232局面を学習する場合ではほぼ同じくらいの時間がかかっており、学習時間あたりの精度向上ではあまり違いが見られなかった。

5. おわりに

本研究では、木カーネルを用いたSVMによる麻雀の打ち手の順位付けの学習を行い、打ち手の順位関係を分類する分類器を作成することで最善手を予測する手法について示した。

結果として、熟練者の打ち手との一致率は51%という値に到達し、それなりに良い精度で最善手を予測できる分類器を得ることができた。特に人手で複雑な特徴要素を作らずとも、簡単な木構造をもとに木カーネルを利用するだけでそれなりの分類が可能であることがわかったと言える。

今後の課題としては、今回問題となった「木構造の適切さ」の検討が必要であると考えられる。順子、対子などの単純な構造のみから木を構築しているために重要な牌を見逃してしまうというケースが見られたので、より柔軟で適切な木構造の構築法を見つける必要がある。例えば面子より大きい4つや5つの牌の塊も構造として見るようにしたり、一つの手牌に対して複数の木構造を対応させるなどの工夫が必要であると考えられる。

また、今回至らなかった「自分の手牌以外の情報の導入」、「他のカーネルや特徴要素との組み合わせ」などの応用を検討することも考えられる。攻めるか降りるかなどの状況判断の知識は現状の木構造のみで表現することは困難なので、分類器以外の仕組みで補うか、別の特徴構造を利用することが必要だと考えられる。手牌は木カーネルを用い、他の情報は線形特徴要素や他のカーネルなどで表現し、それらを組み合わせることでさらに精度を上げることができると期待される。

最後に、今後学習で得られた分類器を評価関数に組み込んで実際にコンピュータゲームプレイヤーを作成し、インターネット上の麻雀対戦サーバで対戦を行ってレーティングを出すなどして、より客観的な評価を得られるようにしていくことが必要であると考えられる。

参考文献

- 1) C.Cortes and V.Vapnik. Support-vector networks. *Machine Learning*, Vol.20, No.3, pp. 273-297, 1995.
- 2) 北川竜平, 三輪誠, 近山隆. 麻雀の牌譜からの打ち手評価関数の学習. *Proceedings of 12th Game Programming Workshop*, 2007.
- 3) Gerald Tesauro. Td-gammon, a self-teaching backgammon program, achieves master-level play. *Neural Comput.*, Vol.6, No.2, pp. 215-219, 1994.
- 4) R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.
- 5) A.Moschitti, D.Pighin, and R.Basili. Semantic Role Labeling via Tree Kernel joint inference. *Proceedings of the 10th Conference on Computational Natural Language Learning*, 2006.
- 6) M.Collins and N.Duffy. New ranking algorithms for parsing and tagging: kernels over discrete structures, and the voted perceptron. *Proceedings of Association for Computational Linguistics*, pp. 263-270, 2002.
- 7) A. Moschitti. TREE KERNELS IN SVM-LIGHT. <http://dit.unitn.it/~moschitt/>.
- 8) A. Moschitti, D. Pighin, and R. Basili. Tree Kernel Engineering for Proposition Ranking. *Proceedings of Mining and Learning with Graphs*, 2006.
- 9) とつげき東北. システムティック麻雀研究所. <http://www.interq.or.jp/snake/totugeki/>.
- 10) T.Joachims. SVM-LIGHT: an implementation of Support Vector Machines (SVMs) in C. <http://svmlight.joachims.org/>.
- 11) T.Joachims. Optimizing search engines using clickthrough data. *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*, ACM, 2002.