

麻雀の牌譜からの打ち手評価関数の学習

北川 竜平[†] 三輪 誠[†] 近山 隆[†]

本研究では不完全情報ゲームである麻雀においてコンピュータプレイヤーの教師あり学習による評価関数のパラメータの調整を行った。評価関数としては3層ニューラルネットワークを用いた。麻雀のゲームの記録である牌譜を用いた学習のために、コンピュータ麻雀プレイヤーが最善手と判断した手と牌譜の中で実際に打たれた手との一致度を測る目的関数を最適制御理論に基づき定義した。目的関数によって求められる値を小さくすることを目的としバックプロパゲーションによって評価関数の調整を行った。結果として、コンピュータ麻雀プレイヤーが最善手と判断した手と牌譜内で打たれた手との一致率はツモ局面において約56%、鳴き局面において約89%であった。インターネット麻雀サーバである東風荘でのレーティングは1318となった。

Learning of evaluation functions in accord with game records in mahjong

RYOUEI KITAGAWA,[†] MAKOTO MIWA[†]
and TAKASHI CHIKAYAMA[†]

In this study, evaluation functions of a computer mahjong player were tuned by supervised learning. The three layer neural networks were used as evaluation functions. For learning by game records, object function based on an optimal control theory was defined for measuring concordance of the best move for computer game player and the selected move in game records. The evaluation functions were tuned by backpropagation for minimization of the value calculated the object function. As a result, a agreement rate of the best move for computer game player and the selected move in game records was about 56% in positions of *tsumo*, about 89% in positions of *naki*. The rating of *Tonpusou* that was internet mahjong server was 1318.

1. はじめに

近年、オセロ・チェス・将棋といった2プレイヤー零和確定完全情報ゲームではコンピュータゲームプレイヤーが人間のトップクラスのプレイヤーと同等以上の実力を持つなどめざましい進歩を遂げている。その一方で不完全情報ゲームでは、ポーカーやコンストラクトブリッジにおいてコンピュータゲームプレイヤーについての研究が行われているが、2プレイヤー零和確定完全情報ゲームと比べるとあまり良い結果は出ていない。不完全情報ゲームでは将来の局面を展開していくといったゲーム木を用いた探索手法を利用することが困難なことがその理由の一つである。不完全情報ゲームは未知の情報が存在する環境において既知の情報を用いることで人工知能に最善の行動をさせることに対する有用なモデルである。よって本研究では未知の情報が存在する環境下における人工知能の学習モデルとして、

教師あり学習を用いることで不完全情報ゲームにおけるコンピュータプレイヤーのパラメータの調整を行う。

本研究では比較的ゲームの記録が入手しやすい麻雀において不完全情報ゲームのコンピュータゲームプレイヤーの作成を行い、教師あり学習によってパラメータの調整を行う。4プレイヤー零和不完全情報ゲームである麻雀は正確なゲーム木の展開が難しいため、従来の探索手法を用いることが困難である。本研究でのコンピュータ麻雀プレイヤーは自分の合法手によって移行が確定する局面のみを評価関数によって評価し、その評価値が最も高いものを最善手として選択する。そのために麻雀の対局の記録である牌譜の打ち手との一致を目的として評価関数の調整を行う。麻雀では評価要素の依存関係によって手の評価が大きく変化することがあるため評価関数としては3層ニューラルネットワークを用いる。評価要素としては自分の手牌、他プレイヤーの捨て牌、場の状態など局面において自分の知ることができる情報を用いる。牌譜を用いた学習を行うために最適制御理論に基づいた目的関数を定義し、その値を小さくすることを目的としてバックプロ

[†] 東京大学大学院新領域創成科学研究科
Graduate School of Frontier Sciences, The University
of Tokyo

パゲーションによって評価関数の調整を行う。

結果として、コンピュータ麻雀プレイヤーが最善手と判断した手と牌譜内で打たれた手との一致率はツモ局面において約 56%、鳴き局面において約 89%となった。またインターネット上の麻雀サーバである東風荘¹⁾で 174 試合戦わせたところレーティングは 1318 となった。

本論文では以降、2 章で関連研究を紹介し、3 章で本研究の手法、4 章で実験結果について説明し、5 章でまとめと今後の課題を述べる。

2. 関連研究

本章では 2.1 で不完全情報ゲームのコンピュータゲームプレイヤーに関する研究について紹介し、2.2 でゲームの記録からの学習によるコンピュータゲームプレイヤーに関する研究について紹介する。そして 2.3 で麻雀における行動選択に関する研究について述べる。

2.1 不完全情報ゲーム

不完全情報ゲームではポーカーやコンストラクトブリッジにおいて研究が行われ、ある程度の強さを持ったコンピュータゲームプレイヤーが作成されている。

D. Billings ら²⁾の作成したポーカープログラム Poki はニューラルネットワークを用い、それぞれの対戦者のとる戦略モデルを構築し学習する。この手法により Poki はインターネット上のポーカーサーバでワールドクラスレベルにまで達している。

小田和ら³⁾のコンストラクトブリッジでは、他プレイヤーの手として構成される可能性のある局面を展開して探索するというアルゴリズムが用いられ、不完全情報ゲームにおいてゲーム木探索を実現している。

2.2 ゲームの記録からの学習

ゲームの記録を利用した教師あり学習は多くのコンピュータゲームプレイヤーに利用され、パラメータの調整において有用な結果をもたらしている。

M. Buro⁴⁾のコンピュータオセロプレイヤーである Logistello は、最小二乗法と最急降下法によりゲームの記録内の与えられた局面からの最終石差をその局面の評価値として学習することで評価関数の調整を行っている。Logistello は人間の世界チャンピオンに勝利といった結果を残している。

D. Gomboc ら⁵⁾は Kendall の順位相関係数を利用して評価関数の調整を行っている。この手法では人間のトップクラスのチェスプレイヤーが局面的有利不利を段階別に分類したものを大量に用意し、コンピュータチェスプレイヤーの探索結果の評価順位と局面的有利不利の順序関係との一致を目的とすることで評

価関数の調整を行っている。

保木⁹⁾によるコンピュータ将棋プログラムである Bonanza は最適制御理論に基づきコンピュータ将棋プレイヤーの探索結果と大量の棋譜の差し手との一致を目的とすることで評価関数の調整を行っている。Bonanza は第 16 回コンピュータ将棋選手権で優勝といった結果を残している。

2.3 麻雀における行動選択

麻雀における行動選択方法としてはとつげき東北⁶⁾⁷⁾の研究が挙げられる。とつげき東北は大量の牌譜を基に統計的手法によってデータの解析を行うことで与えられた局面に対する最適と思われる行動選択方法を提案した。また「流れ」などの古くから信じられてきたオカルトが存在しないことの証明など麻雀研究における土台を作った。

マツタリプログラミング日誌⁸⁾では統計量を人手で組み合わせることで行動選択の評価関数を作成している。その手法によるコンピュータ麻雀プレイヤーであるまったり麻雀は人間のトップクラスのプレイヤーに近い実力に達している。

3. 牌譜を用いた評価関数の学習

本章では牌譜を利用することによる評価関数の学習手法について述べる。まず 3.1 で本研究において使用した麻雀における最善手の選択手法を説明し、3.2 で最善手の決定に必要な評価関数の学習手法について述べる。本研究では保木の最適制御理論に基づいた評価関数の調整を利用し、コンピュータ麻雀プレイヤーが最善手と判断した手と牌譜内で実際に打たれた手との一致を目的として評価関数の調整を行う。

3.1 コンピュータ麻雀プレイヤーの動作の概要

3.1.1 麻雀のルール

本節では基本的な麻雀のルールと用語について説明する。

麻雀は 4 人のプレイヤーが他プレイヤーの点数を奪い合うゲームである。各プレイヤーは 13 枚の牌を手牌とし順に牌を 1 枚引いて (ツモ) 手牌の中から 1 枚捨てる (打牌) という行動を繰り返す。この際手牌は他プレイヤーに見せないため麻雀は不完全情報ゲームである。牌には一から九までの数字のついた牌 (数牌) と文字によって種類が分かれている牌 (字牌) がある。数牌は萬子 (マンズ) ・筒子 (ピンズ) ・索子 (ソーズ) の 3 つの色に分けられる。それぞれの牌は 1 種類につき 4 枚であり牌の枚数は全部で 136 枚である。数牌は同色の 3 連続の数字の牌が同種の 3 枚の牌を集めることで面子 (メンツ) となる。字牌は同種の 3 枚の牌を集める

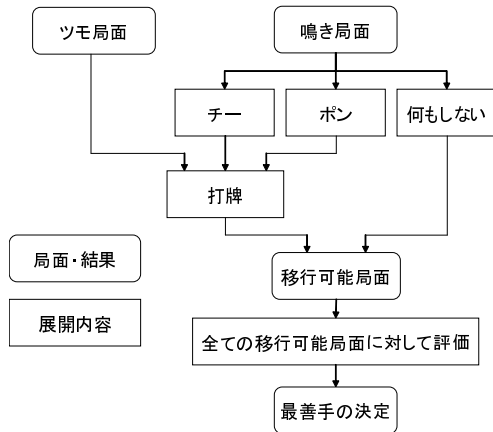


図 1 選択手法の概要

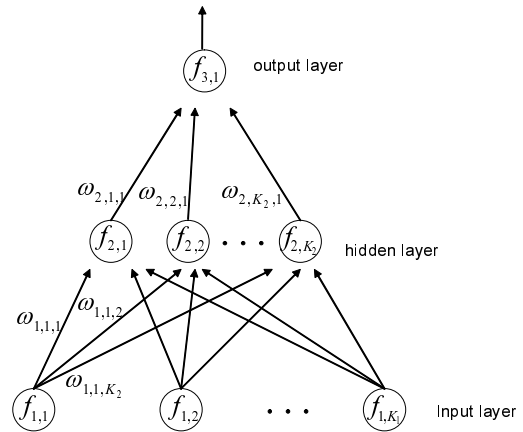


図 2 3層ニューラルネットワーク

ことでしか面子にならない。3連続の数字の牌によって構成される面子を順子（シュンツ）といい、同種の3枚の牌によって構成される面子を刻子（コーツ）という。プレイヤーが打牌した際、他プレイヤーはその牌を手牌に加えることによって面子を構成できるのであれば、それを拾い面子を構成してもよい（鳴き）。鳴きによって順子を作ることをチーといい、刻子を作ることをポンという。また同種の牌4枚を集めたプレイヤーはそれらを一まとめとして面子を構成してもよい（カン）。カンを行ったときのみは4枚で1面子となり、その面子を槓子（カンツ）という。槓子は4枚で3枚の牌とみなされる。カンを行った直後に打牌せずにツモを行うことで手牌の枚数は調整される。ツモによって得た牌か他プレイヤーの捨てた牌と手牌13枚を組み合わせることで4面子1雀頭を構成することができたプレイヤーは上がりとなる。雀頭は同種の牌2枚で構成される。自分の捨てた牌によって上がったプレイヤーは上がったプレイヤーに対し点数を支払わなければならない。またツモで上がったときには他プレイヤー全員が点数を分割して支払う。上がったときの手牌の形によって役が決められており、役の組み合わせにより得られる点数が決まる。この一連の流れを規定回数繰り返し、最終的に最も点数の高かったプレイヤーが勝ちとなる。

3.1.2 コンピュータ麻雀プレイヤー

本研究におけるコンピュータ麻雀プレイヤーでは図1に示すように与えられた局面からの合法手による移行が可能となる全ての局面を展開し、それぞれの局面に対し局面評価をすることで最善手の決定を行う。ここでの局面とは自分の手牌・他プレイヤーの捨て牌・場の状況などを含めた知ることの出来る全ての情報を含めたものを指す。局面評価のために局面から特徴を抽出し評

価関数を用いて評価値を与える。このとき評価値が最大となる局面に移行することのできる合法手を最善手とする。この際他プレイヤーの行動による局面の展開や他プレイヤーの持ち牌の予測などは行わない。この範囲で選択を行うことによって、不完全情報ゲームにおいても確定している情報のみを使用し従来のゲーム木探索に近い手法で最善手を求めることが可能になる。この手法によって作成したコンピュータ麻雀プレイヤーは直接的には役などの麻雀のルールを知らず最善手選択の際にはその知識を用いない。この手法は実質的に深さ1のminmax法であると考えて差し支えは無い。コンピュータ麻雀プレイヤーは牌のツモを行った瞬間の局面（以下、ツモ局面）と鳴くことが可能になった瞬間の局面（以下、鳴き局面）に対してこの手法を用いることで行動選択を行う。ツモ局面における合法手とは「何を切るか」であり、鳴き局面における合法手とは「何もしない」「ポンをして何を切るか」「どの組み合わせでチーをして何を切るか」である。カンにおいては行動後にツモが入り移行局面が確定しないために本研究では選択肢として除外した。

3.2 評価関数の学習

本研究では局面評価のための評価関数に図2のような3層ニューラルネットワークを用い、出力層で得られた値を評価値として用いる。これは麻雀では評価要素の依存関係によって評価値が大きく変化することがあるためである。評価要素としては自分の手牌、他プレイヤーの捨て牌、場の状態など局面において自分の知ることができる情報を用いる。

ニューラルネットワークの重みベクトル $\omega = (\omega_{1,1,1}, \omega_{1,1,2}, \dots, \omega_{2,K_2,1})$ の調整のために牌譜からの教師あり学習を用いる。本研究では与えられた局面

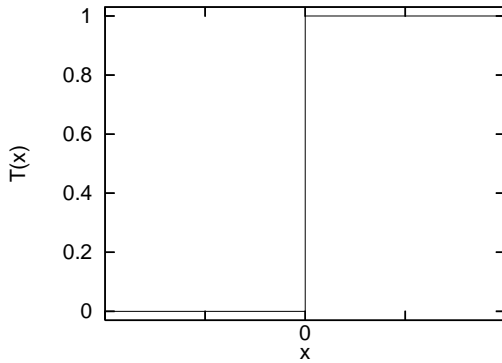


図 3 階段関数

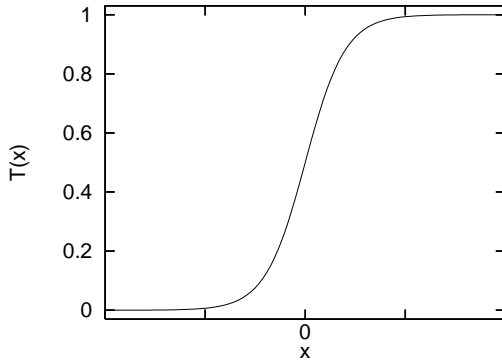


図 4 シグモイド関数

から合法手によって展開される局面のうち最も評価値の高い局面と実際に牌譜で移行した局面とが一致するように学習を行う。

最適制御理論に基づき以下のように目的関数 J を与える。

$$J = \int_0^T l(x, u, t) dt \quad (1)$$

ここで t は時間に関する変数, $x(t)$ は系の状態, u は制御変数である。この目的関数 J の最小化を行うことで最適な制御系を与える。ここで t を学習局面数, $x(t)$ を学習局面, u を重みベクトルとみなすことでコンピュータ麻雀プレイヤーの重みベクトルの調整を行う。

式 (1) より牌譜により与えられた局面に対し離散時間における目的関数 J を

$$J(P_0, P_1, \dots, P_{N-1}, \omega) = \sum_{i=0}^{N-1} l(P_i, \omega) \quad (2)$$

で与える。ここで i 番目 ($0 \leq i \leq N-1$) の局面を P_i とする。 $l(P, \omega)$ は全合法手の評価値の違いの度合いを測る関数であり

$$l(P, \omega) = \sum_{m=1}^M T[V(p_m, \omega) - V(p_0, \omega)] \quad (3)$$

で与える。ここで p_m は局面 P を合法手 m で進めた局面, M を局面 P での合法手の数, $m=0$ を牌譜中で実際に打たれた手, $V(p_m, \omega)$ を局面 p_m の評価値, $T(x)$ を評価値の差を牌譜のプレイヤーとの一致度に変換する関数とする。 $T(x)$ は一価の単調増加関数であり, $|x|$ が大きい範囲での傾きが小さく, $x=0$ 付近で傾きが大きくなる 1 階微分が可能なものを用いる。仮に $T(x)$ を図 3 のような階段型関数にとると, $l(P, \omega)$ は $x > 0$ であるときの局面の数を返す。すなわち $l(P, \omega)$ の計算によって得られる値が牌譜の手よりも良い手と判断した手の数となる。麻雀では全合法手のうち数割が最善手と思われるという局面が頻繁に発生するため, 評価値に差の少ないものであれば得られる値に与える影響も少ないものが望ましい。よって本研究では $T(x)$ として図 4 のようなシグモイド関数を用いる。この目的関数 J によって得られる値が最小となるように重みを調整する。

図 5 に学習方針の概略を示す。学習局面ごとに $l(P, \omega)$ の値が小さくなるように重みを調整することで全学習局面での $l(P, \omega)$ の和である目的関数 J も最小へと近づく。 $l(P, \omega)$ の値を最小とするためにバックプロパゲーションによって重みベクトル ω を調整する。更新前の重みベクトルを ω_{old} , 更新後の重みベクトルを ω_{new} とすると

$$\omega_{new} = \omega_{old} - \eta \Delta \omega_{old} \quad (4)$$

$$\Delta \omega_{old} = \frac{\partial l(P, \omega_{old})}{\partial \omega_{old}} \quad (5)$$

によって重みの更新を行う。このとき η は学習率である。

図 2 のように m 番目の合法手における第 n 層の i 番目のユニットを $f_{n,i,m}$, 第 n 層のユニット数を K_n , 第 n 層の i 番目のユニットから第 $n+1$ 層の j 番目のユニットへの重みを $\omega_{n,i,j}$ とする。各ユニットはシグモイド関数で正規化する。このとき評価値 $V(p_m, \omega)$ は以下のように計算される。

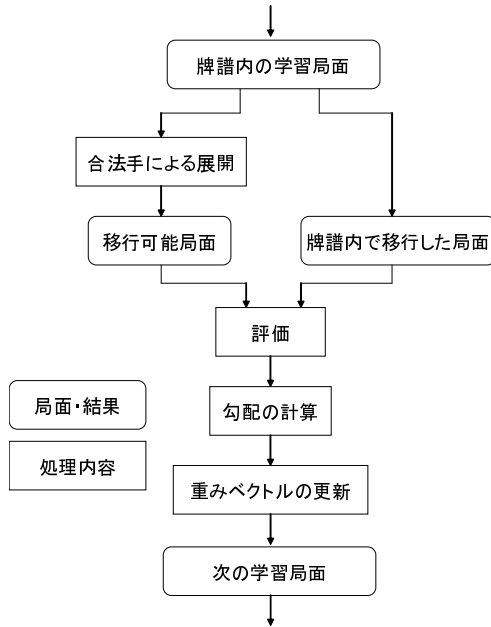


図 5 学習方針の概略

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
	カンチャン搭子とベンチャン搭子の和
	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
他プレイヤーの状態	自分が捨てたことのある牌
	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
場の状態	そのプレイヤーに対する完全安牌
	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 評価要素の概略

$$V(p_m, \omega) = f_{3,1,m} = \text{sigmoid}\left(\sum_{k=1}^{K_2} \omega_{2,k,1} f_{2,k,m}\right) \quad (6)$$

$$f_{2,i,m} = \text{sigmoid}\left(\sum_{k=1}^{K_1} \omega_{1,k,i} f_{1,k,m}\right) \quad (7)$$

ここで $f_{1,i,m}$ には局面 p_m における i 番目の特徴を用いる。

式 (3) (6) (7) より重み $\omega_{n,i,j}$ の調整に使用される勾配 $\Delta\omega_{n,i,j}$ は以下の式で求められる。

$$\Delta\omega_{2,i,1} = \sum_{m=1}^M l(P, \omega)(1 - l(P, \omega))(\Phi_{i,m} - \Phi_{i,m}) \quad (8)$$

$$\Delta\omega_{1,i,j} = \sum_{m=1}^M l(P, \omega)(1 - l(P, \omega))(\Psi_{i,j,m} - \Psi_{i,j,0}) \quad (9)$$

ここで $\Phi_{i,m}$, $\Psi_{i,j,m}$ は以下の式で表される。

$$\Phi_{i,m} = V(p_m, \omega)(1 - V(p_m, \omega))f_{2,i,m} \quad (10)$$

$$\Psi_{i,j,m} = V(p_m, \omega)(1 - V(p_m, \omega))\omega_{2,j,1} \cdot f_{2,j,m}(1 - f_{2,j,m})f_{1,i,m} \quad (11)$$

このように学習を繰り返すことで目的関数 J が小

さくなり、コンピュータ麻雀プレイヤーの行動は牌譜のプレイヤーの行動と似たようなものとなることを期待される。そのため学習対象としてある程度の強さを持ったプレイヤーによる牌譜を用いれば有用な重みベクトルが得られ強いコンピュータ麻雀プレイヤーが作成できると考えられる。

4. 実験

4.1 実験方法

実験は Intel Xeon 3.06GHz dual・メモリ 2GB のマシン上で行った。実装には C++ 言語を用いた。

実験対象としてはシステムティック麻雀研究所で公開されている東風荘の牌譜 6,079 試合のレーティング 2,000 以上のプレイヤーの行動可能局面を用いた。この際対象のプレイヤーのツモ局面と鳴き局面とに分け、それぞれの局面における牌譜内で実際に取られた行動に対し学習を行った。その結果、ツモ局面として 575,906 局面、鳴き局面として 166,348 局面を学習局面として用いた。

表 1 は評価関数に用いた評価要素の概略である。これはゲームの基本的な情報に加えプレイヤーが重視していると思われるものを選択した。本実験では役などの麻雀のルールに関する情報は評価要素としては使用しなかった。これらの評価要素を全て Boolean でしニューラルネットワークの入力ユニットとして使用し

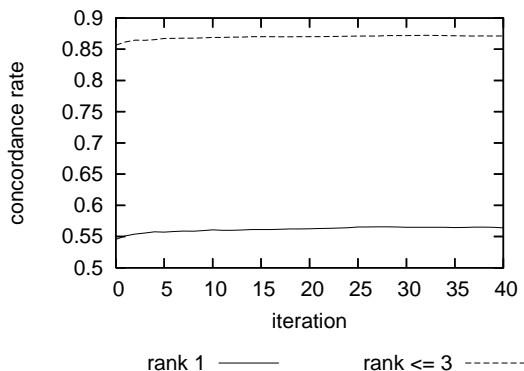


図 6 ツモ局面からの牌譜との一致率

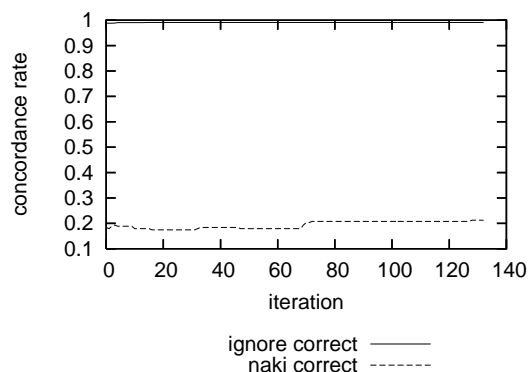


図 8 鳴き局面の正解別による一致率

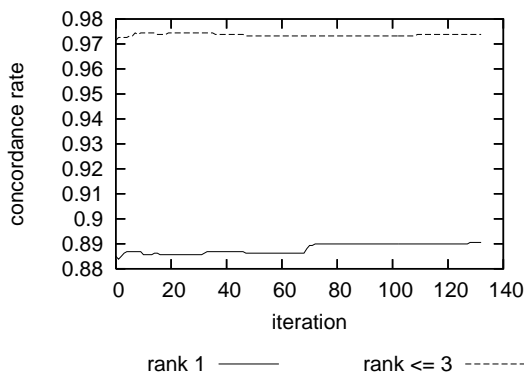


図 7 鳴き可能な局面からの牌譜との一致率

た．その結果入力ユニット数は 1,532 となった．

実験は中間ユニット数を 15，学習率を 0.9×0.99^I として行った．ここで I は学習の繰り返し回数である．このときの学習によって調整するパラメータ数は 22,995 である．この条件による学習を約 50 時間行った．

4.2 実験結果

図 6, 7 にツモ局面と鳴き局面それぞれのコンピュータ麻雀プレイヤーの選択結果と牌譜内で打たれた手との一致率の遷移を示す．図中の実線はコンピュータ麻雀プレイヤーが最善手と判断した手と牌譜の打ち手との一致率であり，点線は牌譜の打たれた手がコンピュータ麻雀プレイヤーの評価順位が 3 位以内となった確率 (以後，3 位以内率) である．ツモ局面において完全一致率は約 56%，3 位以内率は約 87% となった．鳴き局面において完全一致率は約 89%，3 位以内率は約 97% となった．図 8 の実線は正解が鳴きを行わないときの完全一致率であり，点線は正解が鳴きを行うときの

一致率である．鳴きを行わないときの完全一致率は約 99% と高い確率で正解しているのに対し，鳴きを行うときの一致率は約 21% とかなり低い確率となっている．これは学習に用いたデータが鳴かないという判断をするものに偏っているために，多くの状況で鳴かないと判断することにより高い一致率を得ることができてしまうためであると考えられる．

4.3 東風荘における試合結果

実験によって得られた重みベクトル ω を用いてインターネット麻雀サーバである東風荘¹⁾ で試合を行った．実験は Intel Core™ Solo U1400 1.20GHz・メモリ 1GB のマシン上で行った．実装には C++ 言語を用いた．また東風荘での入出力のためにシステムティック麻雀研究所⁶⁾ で公開されている東風荘画面入出力用 DLL を用いた．1 局面の処理 (行動可能状態になってから実際に行動するまで) における平均時間は約 0.2 秒であった．

今回学習対象としていないカン・リーチ・上がりに対する判断は以下の条件に従う．

以下の条件を満たすときのみカンを行う．

- 他家がリーチしていないときの字牌の暗カン
- 他家のリーチが無く自分がリーチしている状態での暗カン

以下の条件を満たさないときの面前テンパイ局面ではリーチを行う．

- リーチをしなくても役があり，満貫以上が確定している
- 待ちが悪形 (カンチャン・ペンチャン・単騎) で役があり，三翻以上が確定している

1 位率	2 位率	3 位率	4 位率	平均順位
0.20	0.27	0.29	0.24	2.575

表 2 各順位の割合

- 待ちが悪形で役もドラも無く、他家がリーチしている
- 残りの上がり牌枚数が 1 以下であり、リーチ以外に一翻以下しかない
- 残りの上がり牌枚数が 0 である
- オールで役があり、上がることで 1 位が確定する
- 4・5・6 のどれかでの単騎待ちである

以下の条件を満たさないときは上がる

- フリテンである
- 役が無い
- オールであり、上がると最下位が確定する

以上の条件と学習により得られた重みベクトルを使い 174 試合行ったところ各順位の割合は表 2 のようになった。またこのコンピュータ麻雀プレイヤーのレーティングは 1318 となった。1300 が東風荘における下位 13%程度のレーティングであるため、このコンピュータ麻雀プレイヤーは弱いと言える。

今回の実験で得た評価関数は面子・両面搭子・対子・ドラの重要性、危険な他プレイヤーに対しては降りるということに関してはうまく学習できた。しかし浮き牌やペンチャン搭子・カンチャン搭子の重要性に関してはうまく学習できなかった。そのために牌効率が悪いと呼ばれる打ち手になってしまっている。また鳴きに関しては役牌を鳴いてテンパイする状況以外ではほとんど鳴きを行わなかった。その他の特徴としてはどのような時でも無難な手を選択するということが挙げられる。例えば自分の手牌が絶望的であり他家のリーチに対し降りるときであっても高い確率で面子を壊さない安全牌や安全の可能性が高い牌を先に切るという選択を行う。これらの問題の多くは評価要素の見直しによって改善できると考えられる。

5. おわりに

本研究では 3 章に示したように牌譜の打ち手との一致を目指しコンピュータ麻雀プレイヤーの評価関数の重みベクトルの調整を行った。具体的には牌譜の打ち手との一致を測る目的関数 J を最適制御理論に基づき定義し、その値が小さくなるように学習を行うことで重みベクトル w の最適化を行った。結果として 4 章に示

したようにツモ局面において約 56%鳴き局面において約 89%の完全一致率を得た。また東風荘で 174 試合行ったときのレーティングは 1318 でありコンピュータ麻雀プレイヤーの実力としては弱いものとなった。

今後の課題としては、1 つは探索に役の知識を入れるということが挙げられる。これを行うには自分がツモや鳴きによって得られる可能性のある牌に対して手牌の局面を展開し、ゲーム木探索を行うことで役への距離を定義し求めるということが考えられる。これによって主に図 8 に示したような正解が鳴きであるときの完全一致率の改善が期待される。

また、もう 1 つの課題としては、カン・リーチ・上がり判断についても学習によって行うといったことが挙げられる。そのためにサポートベクターマシンなどの分類器を用い、それぞれの行動可能局面を「した方がよい」「しない方がよい」の二つに分類を行うことが考えられる。これによって有用なカン・リーチ・上がり判断ができるようになり、レーティングが向上することが期待される。また今回学習対象とした鳴きにおいても行動可能局面において「鳴いた方がよい」「鳴かない方がよい」に分類し、鳴いた方がよいと判断した後にのみ評価関数を用いて行動選択を行うという方法が考えられる。これによって学習対象を実際に鳴いた局面のみとすることができるので、学習データの多くが鳴かない局面であるという偏りによって発生する問題の解決が期待される。

参考文献

- 1) 東風荘 <http://mj.giganet.net/>
- 2) D. Billings, A. Davidson, J. Schaeffer, D. Szafron, The challenge of poker, *Artificial Intelligence* 134, pp. 201–240, 2002
- 3) 小田和友二, 埴敏博, 上原貴夫, コンピュータブリッジのプレイヤーモデルに基づく並列ゲーム木探索, *情報処理学会研究報告* 2006-GI-16, pp. 75–82, 2006
- 4) M. Buro, Experiments with Multi-ProbCut and new high-quality evaluation function for Othello. *Technical Report No.96*, NEC Research Institute, 1997
- 5) D. Gomboc, M. Buro, T. A. Marsland, Tuning evaluation functions by maximizing concordance, *Theoretical Computer Science*, Volume 349, Issue 2, pp. 202–229, 2005
- 6) 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/>
- 7) とつげき東北, 『科学する麻雀』, 講談社現代新書, 2004

- 8) マッタリプログラミング日誌, <http://kmo2.cocolog-nifty.com/prog/>
 - 9) 保木邦仁, Bonanza - The Computer Shogi Program http://www.geocities.jp/bonanza_shogi/
-