牌譜を用いた対戦相手のモデル化とモンテカルロ法による コンピュータ麻雀プレイヤの構築

水上 直紀[1,a)]　鶴岡 慶雅[1,b)]

概要：相手の手や見えない状態を予測することは不完全情報ゲームにおいて重要である．本論文では相手のモデルとモンテカルロ法を用いたコンピュータ麻雀プレイヤの構築法について述べる．相手のモデルは三つの要素（聴牌，待ち牌，得点）の組み合わせとし，各要素を個別に牌譜から予測モデルの学習を行う．モンテカルロ法のシミュレーション中の相手の挙動はこれらのモデルによって得られる確率分布に基づく．オンライン麻雀サイト「天鳳」で作成されたプログラムの実力を評価した結果，レーティングとして，中級者と同等である 1681 点が得られた．

Building computer mahjong players by modeling opponent players using game records and a Monte Carlo method

Naoki Mizukami[1,a)]　Yoshimasa Tsuruoka[1,b)]

Abstract: Predicting opponents' moves and hidden state is important in imperfect information games. This paper describes a method for building a mahjong program that models opponent players and performs Monte Carlo simulation with the models. We decompose an opponent's play into three elements, namely, *tenpai*, *finishing tiles*, and *winning score*, and train predicting models for those elements using game records. Opponents' moves in Monte Carlo simulations are based on the probability distributions of the opponent models. We have evaluated the playing strength of the resulting program on a large online mahjong site "Tenhou". The program has achieved a rating of 1681, which is same as that of the intermediate human player.

1. はじめに

人工知能の分野において不完全情報ゲームは現実世界での応用が期待される非常に挑戦的な研究課題である．不完全情報ゲームにおける最も重要な要素として，相手の行動や見えない状態を予測するということがある．

代表的な不完全情報ゲームのひとつである Texas Hold'em（ポーカーの一種）では，ナッシュ均衡戦略の近似解を事前計算するというアプローチで世界チャンピオンレベルのコンピュータプレイヤが作成された [1] が，プレイヤー数が 3 人以上の場合に 1 ゲームあたりの利益をより増やす試みとして，プレイヤをモデル化して搾取する方法も考えられている．現在の状態を相手のモデル化を利用して推定し，モンテカルロ法によって，利益を算出する方法 [2] や相手の挙動から手の強さを推定し，木探索によって期待値を計算することで搾取を行う方法 [3] が提案されている．

相手の手を予測する方法として，Skat というゲームでは，相手の手を予測する方法として，手を選んだ時の特定の場の状況である確率を棋譜から算出するという手法が提案されている [4]．確率値は，ベイズの定理を用いて場の状況から計算される．平均プレイヤの棋譜から学習した結果，エキスパートレベルの実力を持つコンピュータプレイヤが得られたと報告されている．

本論文が対象とする「麻雀」は，4 人零和不確定不完全情報ゲームである．コンピュータ麻雀プレイヤに関する研究としては，一人麻雀を定義して四人麻雀に足りない機能を拡張する方法 [5] や牌譜から危険牌予測を行う方法 [6]，やモンテカルロ木探索を適応する方法 [7] などに関する研究がある．

本論文で提案する麻雀プレイヤの実現法の概要について

[1] 東京大学
The University of Tokyo
[a)] mizukami@logos.t.u-tokyo.ac.jp
[b)] tsuruoka@logos.t.u-tokyo.ac.jp

説明する．相手のモデル化として相手の具体的な手牌を直接予測するのではなく，聴牌しているか，待ち牌は何か，得点は何点かといった三つの要素として抽象化したモデル化を行う．まず，牌譜からプレイヤの聴牌予測，待ち牌予測，得点予測について各モデルの学習と予測精度について人間との比較を行う．次に，得られた学習器とモンテカルロ法を用いた手の決定方法について，シミュレーション中の挙動を決定して，牌譜との一致率をもとにパラメータのチューニングを行う．最後に，得られたプレイヤをオンライン麻雀サイトでの人間との対戦やほかのソフトとに対戦によって評価する．

本論文は以下の構成になっている．初めに 2 章で麻雀のルールと用語，3 章で関連研究を述べる．提案手法として，4 章で相手プレイヤの聴牌予測，5 章で待ち牌予測，6 章で得点予測，7 章でモデル化とモンテカルロ法を用いた手の決定，8 章で提案手法の対戦結果について述べる．最後に 9 章でまとめについて述べる．

2. 麻雀のルールと用語

この章では麻雀のルールと用語について解説する．麻雀は自分の牌を組み合わせて役（特定の構成）を作り，**和了（ホーラ）**し，役に応じた点数を得るゲームである．和了するため，自分の牌を**ツモ**と呼ぶ 1 枚牌を引く行為と 1 枚牌を捨てる行為と**鳴き**または**副露（フーロ）**と呼ぶ，捨てた牌を利用する行為を各プレイヤが順番に行う．和了に関する用語としては，ツモして和了することもツモと呼び，相手の捨てた牌で和了することを**ロン**と呼ぶ．またロンされることを**放銃**と呼ぶ．

牌は全部で 34 種類あり，それぞれが 4 枚あるため合計は 136 枚となる．牌には数字の 1〜9 のいずれかが書かれた**数牌**と文字の書かれた**字牌**がある．数牌は 3 種類あり優劣はない．字牌は 7 種類あり，東，南，西，北の総称である風牌と白，発，中の総称である三元牌に分けられる．

和了に必要な 14 枚の組み合わせを完成させるためには，**面子（メンツ）**と呼ばれる特定の 3 枚の組み合わせが 4 組と，**対子（トイツ）**と呼ばれる 2 枚の同一牌が 1 組必要である．メンツは同じ牌を 3 つ集める**刻子（コウツ）**と数字の連続する 3 つ牌を集める**順子（シュンツ）**がある．

また以下に一般的な麻雀用語を説明する．

リーチ　鳴きを一度も行わず，あと 1 枚で和了できる手牌になった時宣言出来る行為である．リーチ宣言の後は手牌を変更することはできない．すなわち，リーチ後にツモで入手した牌は，和了できる場合を除きすべてそのまま捨てなくてはならない．

聴牌（テンパイ）　和了に必要な牌の枚数が 1 枚の状態

役牌　3 つ集めると役になる牌

オタ風牌　役牌でない風牌

ドラ　1 局の始めにランダムに決定する牌．和了した時にこの牌があれば得点が増加する

山　同じ配牌，ツモになる牌の並び

降り　自分の和了を諦め，相手の和了牌を捨てないようにする行為

回し打ち　和了を目指した最善手を選ぶのではなく，放銃を避けつつ和了や聴牌を狙う行為

現物　相手がすでに捨てた牌．ルール上，現物をロンされることはない．

筋　現物の数字の 3 つ外側にある牌

壁　数字の連結がなくなった牌

染め　自分の手牌を同じ色で揃える役（ホンイツ，チンイツ）の総称

タンヤオ　各色の 1，9 牌と字牌を一枚も使わない役

トイトイ　メンツがすべて刻子で構成された役

3. 関連研究

麻雀の研究としては以下の研究が報告されている．水上ら [5] は一人麻雀を定義して牌譜をもとに学習を行い，四人麻雀との牌譜との差異を見つけ，埋めることで四人麻雀の実現させた．これにより麻雀の実力向上に必要な部分は降りと鳴きであることが分かり，改善した結果，平均プレイヤ以上の実力を得ることが出来た．降りを実現させるためには，牌譜に降りるべき局面を人手でタグをつけることが必要である．しかし自分の手牌の良さや場の状況の組み合わせが多いため，学習に必要なデータを大量に集めることが困難であるという問題がある．またこの方法では回し打ちが出来ないことも問題である．

捨て牌の危険度の推定を推定する研究として我妻らは SVR を用いた方法 [6] を提案している．これは牌譜に記録された局面データから特徴ベクトルを抽出して危険度として期待損失点の予測を行うものである．しかし教師データに聴牌している局面が少なく学習が上手くいかない問題や人間との一致度という主観的な方法で推定の精度を求めなければならない問題があった．

三木らはモンテカルロ木探索を用いた麻雀プレイヤを提案している [7]．この手法では，麻雀のゲーム木の探索は探索の分岐数が膨大であり，正確に行うのは困難なため，相手の手牌や行動をランダムでシミュレートするモンテカルロ木探索を用いている．麻雀の知識をほとんど使わないにもかかわらず，和了に必要な牌の枚数を下げるように打つという単純なルールに基づいたプレイヤよりも成績が上回る結果となった．しかし，相手はシミュレート時にほとんど和了できないため，役を作ることが困難になる鳴きをするというように，人間が行うプレイとは大きく異なる挙動を示した．

モンテカルロ木探索が成功した例として，コンピュータ囲碁の世界では Crazy Stone [8] がヒューリスティックに基づいた確率分布による手をシミュレーション中に選択することで 9 路盤であれば，プロ並みの実力を得ている．

ポーカーのモデル化の研究としては以下の研究が報告されている．あらかじめモデルの状態数と状況においての行動の頻度を計算しておく方法として，ゲームの棋譜を集め，

今までの行動履歴や頻度，場の状況から複数の相手モデルを作り，モンテカルロ法を用いることで適応する方法 [2] が提案されている．

ゲーム中に相手モデルを更新して搾取する方法 [3] もある．相手の手が開かれるたびにそのゲームでの行動をもとにその状況での手の強さの頻度分布を更新し，Expectimax 探索行うことで自分の手の決定を行うものである．

これらのポーカーの研究は相手が固定されているが，麻雀の場合，インターネット麻雀サイトなどでは不特定の相手と対戦であり，対戦数も 1 ゲームのみであるため一人一人モデル化を行うのは困難である．また特定の相手と連続して対戦する場合にしても相手の手牌を具体的に予測することが人間にも難しいため，行動を予測するのは困難である．また人間であっても一人の相手に着目して搾取を行うなどは戦術として語られていないことからも現実的とは言えない．

4. 相手プレイヤの聴牌予測

この章では相手が聴牌しているかどうかを予測するモデルを学習する．麻雀のルール上，聴牌していなければ，ほかのプレイヤに点数に関して直接影響を与えることは不可能である．ゲームを人間が行う時には自分の聴牌，和了を目指し，相手も同じように聴牌，和了を目指すと考えるので，相手が聴牌をしているかどうかの判断は重要である．

4.1 聴牌予測の学習

聴牌予測には膨大な特徴量が必要であり，その調整を高速に行う必要がある．そこで聴牌予測には牌譜における聴牌かどうかの一致を目指した，2 値のロジスティック回帰モデルを用いる．

$\mathbf{x_p}$ を局面の相手プレイヤ p に対する特徴ベクトル，$\mathbf{w}$ を重みベクトルとすると，ある局面の聴牌率 $Pr(tenpai)$ は式 (1) で計算する．

$$Pr(tenpai) = \frac{1}{1 + \exp\left(\sum_{i=1}^{n} -\mathbf{x_{p,i}} \mathbf{w_i}\right)} \tag{1}$$

$\mathbf{X_N}$ を $\mathbf{x}$ のデータ集合，そのラベル値を $\mathbf{c_N}$，λ を正則化項とすると，目的関数は式（2）で表される．学習はこの目的関数を最小にする $\mathbf{w}$ を求めることである．

$$L(\mathbf{w}) = -\sum_{i=1}^{N} (\mathbf{c_n} Pr(\mathbf{X_i}) + (1 - \mathbf{c_n})(1 - Pr(\mathbf{X_i})) + \frac{\lambda \mathbf{w^T w}}{N} \tag{2}$$

この重みベクトルの学習は FOBOS [9] を用いて学習を行う．学習率 η は Adagrad [10] を用いて決定する．Adagrad [10] は重みベクトルが更新されるたびに学習率が小さくなっていくアルゴリズムであり，式 (3) で表現される．添え字の i は特徴ベクトルの i 番目の要素，t は t 回目の更新時の値を示している．

$$\mathbf{w_{t+1,i}} = \mathbf{w_{t,i}} - \frac{\eta g_{t,i}}{\sqrt{1 + \sum_{k=1}^{t} g_{k,i}^2}} \tag{3}$$

特徴量	次元数
リーチを打っているか	1
副露数と捨て牌の数	$5 \times 19 = 95$
副露数とその順目	$4 \times 19 = 76$
副露数と手出しの数	$5 \times 19 = 95$
副露数と最後に手出しした牌の種類	$5 \times 37 = 185$
副露した種類と副露した時に切った牌の種類	$136 \times 37 = 5032$
ドラの種類を切ったか	34
赤ドラを切ったか	1
手出し牌とその次の手出し牌の組み合わせ	$37 \times 37 = 1369$

表 1 聴牌予測の特徴量

プレイヤ	AUC
上級者	0.778
分類器	0.777

表 2 聴牌に関する評価

特徴量を **表 1** に示す．特徴ベクトルの次元は 6,888 になった．

4.2 学習に用いる牌譜

上記の聴牌予測には多くの教師データが必要になる．教師データとしてはインターネット麻雀サイト天鳳 [11] の鳳凰卓の牌譜を用いた [*1]．鳳凰卓でプレイできるのは全プレイヤの中でも上位 0.1% 程度であり牌譜の質は高いと考えられる．以下，牌譜と示した場合はこの鳳凰卓の牌譜のことを指す．

牌譜中のプレイヤの手番においてプレイヤの聴牌かどうかの 2 値データと，そのプレイヤ以外のプレイヤの視点から観測できる特徴量を生成し，教師データとする．局面数はおよそ 1.77×10^7 である．

4.3 聴牌予測の精度

学習が上手くできているかを調べるため，得られた分類器の聴牌予測の精度を調べた．評価は ROC 曲線下面積（AUC）を用いる．テストデータは牌譜から 1 局の中で 1 局面に限定し，100 局面を選択した．リーチしたプレイヤが聴牌しているかを予測するのは容易であるため，このテストデータにこのような状況は入っていない．学習率は 0.01，正則化項は 0.01 を使用した．

次にテストデータに対しての結果を **表 2** に示す．分類器は聴牌を判断する能力においては上級者に近い実力を示めしている．上級者は麻雀サイト天鳳 [11] において鳳凰卓でプレイすることができるプレイヤのことを指す．これは全プレイヤの中でも 0.1% ほどであり上級者といって差し支えないであろう．

5. 待ち牌の予測

この章では相手が聴牌しているか時の相手の待ち牌を予測するモデルを学習する．麻雀のルール上，相手の待ち牌を自分が切ってしまった場合，相手の得点をすべて自分が払わなければならない．そのため相手の待ち牌を高精度で

[*1] 2009 年 2 月 20 日から 2013 年 12 月 31 日までに行われた対局

特徴量	次元数
牌の種類とその牌が何枚見えているか	$5 \times 34 = 170$
現物かどうか	34
n 回（n=0，1，2）手出しの間に通った牌	$3 \times 34 = 102$
ドラの種類	34
順目とその時に捨てた牌	$18 \times 37 = 666$
リーチ時に切った牌	37
副露の種類と副露時に切った牌	$136 * 37 = 5032$
二つの副露の種類	$136 * 136 = 18496$
捨て牌二つの種類と手出しかどうか	$37 \times 37 \times 2 \times 2 = 5476$
手出し牌とその次の手出し牌の組み合わせ	$37 \times 37 = 1369$

表 3 待ち牌予測の特徴量

予測可能になることは重要である．

5.1 待ち牌の予測の学習

待ち牌は一般的には複数あるため，教師データは麻雀の牌の種類である 34 種類の待ち牌かどうかの 2 値分類の形になっている．それぞれの牌の種類について 2 値のロジスティック回帰モデルを用いて学習を行う．

目的関数は式（4）で表される．

$$L(\mathbf{w}) = -\sum_{i=1}^{N}(\mathbf{c_n}Pr(\mathbf{X_i}) + (1-\mathbf{c_n})(1-Pr(\mathbf{X_i})) + \frac{\lambda|\mathbf{w}|}{N} \tag{4}$$

この重みベクトルの学習は前述の FOBOS [9] と Adagrad [10] を用いて学習を行う．

特徴量を**表 3** に示す．特徴ベクトルの次元は 31,416 になった．

5.2 学習に用いる牌譜

上記の待ち牌の予測の学習には多くの教師データが必要になる．牌譜中のプレイヤの手番においてプレイヤの待ち牌とそのプレイヤ以外のプレイヤの視点から観測できる特徴量を生成し，教師データとする．局面数はおよそ 7.24×10^7 である．

5.3 待ち牌の予測の精度

待ち牌の予測の評価は麻雀のルールの特徴を反映している必要がある．麻雀のルール上，相手の待ち牌を一枚でも切った時点で終局するため，複数の待ち牌のうち一つだけが高精度で予測可能であっても意味がない．そこで評価としては次の手順を踏む．ある局面において特定のプレイヤに対し，待ち牌の確率が低いと予測した順に自分の牌を並べる．次にその順番通りに牌を切ったとして，初めて待ち牌と一致するまでの回数を数える．その数と和了しない牌の種類（理想値）で割った値を用いる．複数の局面をテストする場合は一致までの回数と理想値の総和を割った値を用いる．テストデータは牌譜から 1 局の中で 1 局面に限定し，100 局面を選択した．なおテストでは局面に複数人が聴牌していても予測を行うプレイヤは 1 人に限定し，テストする被験者にはそのプレイヤの待ち牌が自分の手牌の中にあることを伝えている．

プレイヤ	評価値
上級者	0.744
分類器	0.676
ランダム	0.502

表 4 待ち牌予測に関する評価

学習率は 0.01，正則化項は 0.01 を使用した．テストデータに対する結果を**表 4** に示す．ランダムとは待ち牌の順番をランダムに並べ替えることである．分類器はランダムより性能が高く，上級者よりも低い結果になった．

6. 得点の予測

この章では相手に対して待ち牌を切ってロンされた時の支払う得点を予測するモデルを学習する．この章はロンが主だが，ツモに関しても同様に学習を行う．

6.1 得点の予測の学習

麻雀のルール上，得点は翻数の 2 のべき乗に基本的には比例して増えていく．そのためゲームの点数をそのまま使用して学習を行った場合，同じ翻数の誤差であっても大きい翻数の方が得点の誤差が大きくなる．そこで教師データを作る際に得点に自然対数をとった．また点数は親が上がった時でも子の点数に変換を行った．教師データに対して重回帰モデルで学習を行う．

相手の予想得点 $Tokuten$ は式 (5) で計算する．

$$Tokuten = \sum_{k=1}^{i}\mathbf{x_k}\mathbf{w_k} \tag{5}$$

目的関数は式（6）で表される．

$$L(\mathbf{w}) = \sum_{i=1}^{N}(Tokuten - \mathbf{c_n})^2 + \frac{\lambda\mathbf{w^T}\mathbf{w}}{N} \tag{6}$$

この重みベクトルの学習は前述の FOBOS [9] と Adagrad [10] を用いて学習を行う．

重回帰モデルの計算は最小二乗法を用いて一意の解を求めるが可能であるが，この学習では学習の局面が多く計算機のメモリに乗らず，計算することが不可能である．そこで前述の FOBOS [9] と Adagrad [10] を用いて学習を行う．

特徴量を**表 5** に示す．特徴ベクトルの次元は 26,889 になった．

6.2 学習に用いる牌譜

上記の得点予測の学習には多くの教師データが必要になる．牌譜中のプレイヤの手番においてプレイヤの待ち牌を切られた時の得点と，そのプレイヤ以外のプレイヤの視点から観測できる特徴量を生成し，教師データとする．局面数はおよそ 5.92×10^7 である．

6.3 得点の予測の精度

評価は予測した得点と実際の得点との平均二乗誤差を用いる．テストデータは牌譜から 1 局の中で 1 局面に限定

特徴量	次元数
親かどうかとリーチしているかどうか	$2 \times 2 = 4$
確定している役と見えているドラの枚数	$7 \times 8 = 56$
リーチかダマか副露と確定している役と見えているドラの枚数	$3 \times 7 \times 8 = 168$
副露の種類と確定している役と見えているドラの枚数	$136 \times 7 \times 8 = 7616$
二つの副露の種類	$136 \times 136 = 18496$
副露数と確定している役と見えているドラの枚数	$5 \times 7 \times 8 = 280$
リーチしているかどうかか切った牌が筋になっているかタンヤオ牌かどうか	$3 \times 2 \times 2 = 12$
オタ風を鳴いた，さらに役牌を鳴いた，何も鳴いていない	3
切った牌がドラ，ドラの一つ隣，二つ隣，同じ色，無関係	5
タンヤオが可能な副露とドラがタンヤオと見えてるドラの枚数	$2 \times 2 \times 8 = 32$
ホンイツが可能な副露と確定している役とドラが染め色かどうか	$5 \times 7 \times 2 = 70$
チンイツが可能な副露と確定している役とドラが染め色かどうか	$5 \times 7 \times 2 = 70$
トイトイが可能な副露と確定している役とドラが染め色かどうか	$5 \times 7 \times 2 = 70$
三元牌が何種類鳴かれているか	3
風牌が何種類鳴かれているか	4

表 5　得点予測の特徴量

プレイヤ	評価値
上級者	0.40
分類器	0.37

表 6　得点予測に関する評価

し，100 局面を選択した．なおテストでは被験者には何の牌を切り，どのプレイヤが上がるかを伝えている．

学習率は 0.01，正則化項は 0.01 を使用した．テストデータに対しての結果を **表 6** に示す．分類器は上級者を超える実力を得た．

7. モデル化とモンテカルロ法を用いた手の決定

前章までの結果から相手の抽象化したモデル化は上級者並みの精度で行えることを示した．この章ではこの抽象化したモデルを用いて自分の手の決定を行う．抽象化したモデルにより相手の得点を予測することが出来るが，一人麻雀の手の決定アルゴリズムにおいては，自分の手牌を良さを表す評価値はゲーム上の得点とは異なる．そこでモンテカルロ法を用いて自分の手牌をゲーム上の得点として扱うことにする．式（7）は手を選択するために用いる式であり，ある牌を Hai としたときのゲーム上の得点を計算する．$Sim(Hai)$ はこれから説明するモンテカルロ法によって求められるゲーム上の得点であり，二つ目の項はこの牌を切った時の期待損失点である．相手が親の時は期待損失点は 1.5 倍する．**図 1** に示すように，シミュレーションは自分の牌だけでなく，後述の “仮想的降り” を行った時の得点（$Fold$）も計算する．$Sim(Hai)$ はシミュレーションの値と $Fold$ と比較して大きい値とする．最終的に各牌についての $Score$ を求め最大のものを実際のゲームで切る牌とする．なお本論文では “一人麻雀の手” というのは水上らが行った手法 [5] の鳴くことが可能で降りを行わない和了を目指した手のことを指す．また “モンテカルロ法の手” はシミュレーションによって選ばれる手である．

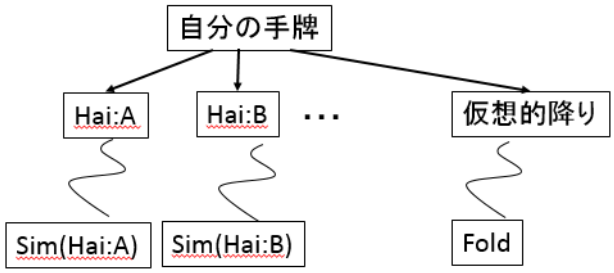

図 1　モンテカルロ法を用いた手の決定の概要

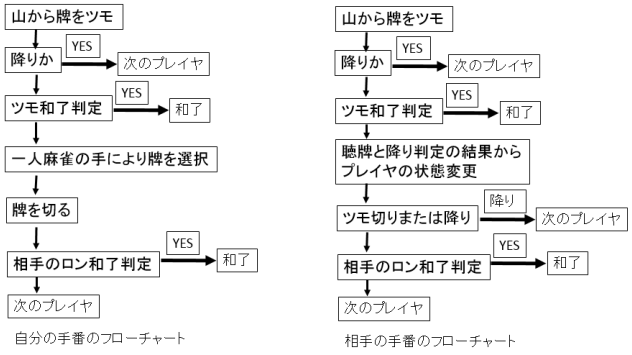

図 2　各プレイヤのフローチャート

$$
\begin{aligned}
Score(Hai) &= Sim(Hai) - \sum_{k=1}^{3} EV(k, Hai) \\
EV(k, Hai) &= Pr(k = Tenpai) \times Pr(k, Hai = Agari) \times Tokuten(k, Hai) \\
Sim(Hai) &= Max(Sim(Hai), Fold)
\end{aligned}
\tag{7}
$$

麻雀におけるモンテカルロのシミュレーションには自分と相手の手番と山であるチャンスノードの時の挙動を決める必要がある．山は自分の見えている牌を数え，それ以外の牌をランダムに配置することで決定する．**図 2** は自分と相手の手番のフローチャートである．自分の手番においては一人麻雀の手を選択する．相手の手番は抽象化したモデルが確率に基づいてリーチや和了また仮想的降りを行う．切られる牌については山からのツモ牌をツモ切りすること

で決定する．このシミュレーションは一局終了まで行う．シミュレーション回数は全ての手に対して同じ回数だけ行う．シミュレーションは全ての手に対して山や相手の挙動を決める乱数は運の影響を少なくするため同じ数字を用いる．またモンテカロ木探索のようにゲーム木を深くすることは行わない．図 2 の降りは仮想的降りを指す．次の節から具体的に記述する．

7.1 自分の手番

自分の手牌のすべての牌に対して，その牌を切った局面からシミュレーションを行う．また鳴きが可能な局面では鳴いて牌を切った局面と鳴かない局面からシミュレーションを行う．

仮に現局面が降りるべき局面の時，その子ノードでは降りるための牌を切ることが出来るものの，シミュレーションでは和了しか目指さないため降りの戦略をとり続けた時の得点を概算できない．そこで仮想的降りを導入し，現局面から一局終了まで降りの戦略を続けた時の得点を求める．仮想的降りは図 2 にあるように，山から牌をツモリ，山の数を減らすものの何も切らないで次のプレイヤの手番とする行動である．このシミュレーションでは放銃はしないが，ツモられた時の得点と流局時の得点を払うことになる．

7.2 相手の手番

相手の手番では相手の具体的な手牌は手を決定せずにモデル化した確率分布に基づいて行動する．シミュレーション中に相手が和了するには相手が聴牌であり，かつ切られた牌やツモ牌が和了牌である必要がある．切られた牌やツモ牌が和了牌であるかの判定を行うためには現局面での待ち牌予測した確率を用いる．シミュレーション中に待ち牌の確率分布は更新しない．

シミュレーション中の相手の聴牌かどうかを決定するには，相手のツモ局面において判定行う．聴牌していないプレイヤが聴牌する確率は一人麻雀の手を用いてあらかじめ調査した．牌譜を使用しないのは，牌譜には降りが含まれており，聴牌率が実際よりも低くなるためである．一人麻雀プレイヤに配牌とツモを与え，各順目において聴牌していない局面から聴牌になった局面を調べ，各順目の聴牌率を求めた．相手の聴牌の種類としては鳴きとリーチの二種類ある．リーチと鳴きの各順目の聴牌率を調べるときに，前者では相手の捨て牌がなく，後者は捨て牌があり，牌を鳴くことが出来る．それぞれ 10^6 回行って各順目の聴牌率を調べた．結果を **図 3** に示す．

相手の聴牌しているかの初期値は，シミュレーション開始時に予測した聴牌率を用いて決定する．鳴きとリーチのうちどちらで聴牌するかはシミュレーション開始時に決定する．その方法は現局面で副露またはリーチしていないプレイヤに対して，副露率をもとに決定する．副露率は鳳凰卓の平均である 35%とした．

このシミュレーション中での相手の仮想的降りについて

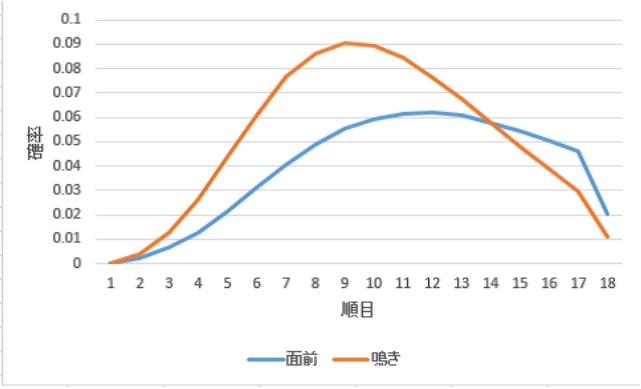

図 3 各順目での初めて聴牌する確率

の説明を行う．仮に降りがなければ，自分やほかの相手がリーチをしている局面などではリーチしたプレイヤが和了しやすくシミュレーションとしては適切でない．そこで自分の手番で前述した仮想的降りを用いる．自分の手番と相手の手番での仮想的降りとの差異は，自分の手番での仮想的降りは最初から最後まで一局を通して降りであるが，相手ノードでの仮想的降りはシミュレーションの途中からでも降りが可能なことである．

シミュレーション中に仮想的降りを行うかどうか判定は相手の手番の開始時に，場の状況からプレイヤが降りる確率を用いて判定する．一人麻雀の手と牌譜と一致しない局面は，降りた局面が多いこと [5] を利用して，降りる確率を牌譜と一人麻雀の手を用いて調べる．牌譜における自分が聴牌しているかという情報と相手の副露数とリーチしているプレイヤの数を調べ，場の状況として，その局面において一人麻雀プレイヤの選択する上位 3 つの中に牌譜で切られた手がなかった割合をその場の状況での降りる確率とする．例えば自分が聴牌しておらず，リーチしたプレイヤが一人であるときは一致率が 60%である．一度仮想的降りの状態になるとその局はすべて降りを行う．仮想的降りの状態ではロン和了することはない．

7.3 手の決定

序盤では自分や相手の和了までに時間がかかり手の評価が短い時間ではモンテカルロ法の手は精度が悪いことが予想される．そこで一人麻雀の手とモンテカルロ法の手を使い分ける方法を考える．その方法は序盤では一人麻雀の手を用いて，予測した相手の聴牌率が一人でも閾値を超えた時にはモンテカルロ法の手を採用する．この閾値は牌譜との一致率を用いて決定する．

牌譜の局面数は 10,000 局面，モンテカルロ法の手にかける時間は一手 1 秒とした．結果を **図 4** に示す．Rank n は第 n 候補に牌譜での打牌が入っている割合とする．閾値が 0 の時はすべて一人麻雀の手であり，閾値が 1 の時はすべてモンテカルロ法の手であり，これらを閾値で分離することで牌譜一致率が上がる．実験の結果から閾値は 0.9 を用いて手を決定する．

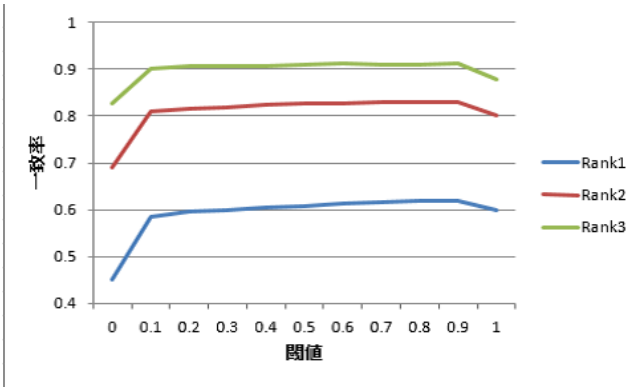

図 4　各閾値における牌譜一致率

	1 位率	2 位率	3 位率	4 位率	平均順位
まったり麻雀	0.248	0.247	0.250	0.255	2.51
提案手法	0.230	0.248	0.268	0.254	2.54
水上	0.243	0.226	0.222	0.309	2.59

表 7　対まったり麻雀での順位分布

8. 結果

提案手法によって得られた麻雀プログラムをほかの麻雀プログラムであるまったり麻雀 [12] とインターネット上の麻雀対戦サイトである天鳳 [11] で対戦させた．

8.1 まったり麻雀との対戦における評価設定

実力を測定するため，まったり麻雀 [12] と対戦を行った．まったり麻雀 [12] は統計量をヒューリスティックに組み合わせて行動選択の評価関数を作成している．その実力は筆者の知る限り現存する麻雀プログラムの中で一番高い．

ルールは東風戦，赤あり，ゲーム自体に制限時間はないが，一手 1 秒で行った．対戦相手の内訳はまったり麻雀が 3 人と提案手法のプログラムが 1 人である．対戦方法としてまったり麻雀の対戦モードの中のデュプリケートモードを用いた．これは最初に山を作成するときに使用する乱数を選択できるモードで，これにより配牌やツモについての運の要素をなくすことができる．またオプションにより同じ乱数でまったり麻雀自体が打った成績と比較することができる．乱数は 0〜999 を用いて 1,000 試合を行った．評価は平均順位を用いる．

8.2 まったり麻雀との対戦における結果

提案手法と以前の水上らの手法 [5] とまったり麻雀 [12] の 3 つの比較を行った．結果を**表 7** に示す．まったり麻雀に勝ち越すことやまったり麻雀自体が出した成績を超えることはできなかったが，以前の研究よりも 4 位率が大きく改善され，平均順位が良くなっており実力が向上したといえる．

和了・放銃率は **表 8** に示す．提案手法はまったり麻雀と同じ成績を出すことができた．しかしながら平均順位で負けているのは今の順位や得点によって戦略を変更できない方法にあると考えられる．

	和了率	放銃率
まったり麻雀	0.200	0.122
提案手法	0.201	0.121
水上 [5]	0.228	0.178

表 8　対まったり麻雀での和了・放銃率

	和了率	放銃率
提案手法	0.237	0.127
水上 [5]	0.256	0.148

表 10　和了・放銃率

8.3 天鳳での対戦における評価設定

実力を測定するため，インターネット上の麻雀対戦サイトである天鳳 [11] で対戦を行った．ルールは東風戦，赤あり，持ち時間は一手 3 秒に考慮時間が 5 秒である．提案手法のプログラムは一手 1 秒で行った．鳴きについても同様の持ち時間があり，鳴ける局面では次のプレイヤは勝手に牌を引くことができない．天鳳 [11] では成績に応じて対戦できる卓が異なる．卓は 4 種類あり，上から鳳凰卓，特上卓，上卓，一般卓がある．卓の種類は上の卓を選択可能な成績であっても上級卓のみに限定した．プログラムをサイト上で対戦させるための入出力インターフェースは自作した．評価としては平均順位（レーティング）を用いる．レーティング（R）とは平均順位と負の相関を持つ基準である．具体的には（8）で計算される．

$$R' = R + (50 - Rank \times 20 + \frac{AveR - R}{40}) \times 0.2 \quad (8)$$

$Rank$ は前回のゲームでの順位である．$AveR$ は卓の平均の R である．初期 R は 1500 であり，およそ平均順位が 0.1 下がるごとにレーティングは 100 点ほど上昇する．

レーティングを用いた評価方法としては安定レーティングと保証安定レーティング [13] の二つが提案されている．安定レーティングは今までの順位分布をとり続けた場合のレーティングである．保証安定レーティングは連続する試合の結果を恣意的に切り出した時の安定レーティングを用いて，安定レーティングの現実的な範囲を保証する安定レーティングのことである．これは異なる試合数の恣意的に良かった結果を取り出しても比較可能な評価方法である．

8.4 天鳳での対戦における結果

提案手法と以前の水上らの手法 [5] の 2 つの比較を行った．対戦結果を**表 9** に示す．安定レーティングは大きく変わらないが，保障安定レーティングの向上が見られる．

和了・放銃率は **表 10** に示す．結果としてよくなったとは言えないが，提案手法の和了率と放銃率はより人間の平均値に近い値を出した．

8.5 考察

提案手法の挙動を見る限り，自分の和了が遠く相手が

	1 位率	2 位率	3 位率	4 位率	平均順位	試合数	安定レーティング	保障安定レーティング
提案手法	0.231	0.269	0.268	0.233	2.50	2227	1681	1690
水上 [5]	0.253	0.248	0.251	0.248	2.49	1441	1689	1610

表 9 順位分布

リーチしている局面においては降りになる牌を選択することが多い．また回し打ちの挙動も見られるなど人間に近い挙動を確認した．

まったり麻雀 [12] を相手にしたときでは成績が向上したのに対して，天鳳 [11] で対人戦を行ったときはあまり成績が向上しなかった．その原因としては，まったり麻雀 [12] の牌効率が上級卓のプレイヤより上手く，聴牌率が高いためモンテカルロ法を用いることが多かったためと考えられる．

問題として，モンテカルロ法を用いる閾値を相手の聴牌率のみで判定していたため，相手が副露して高い手を作っている局面において，一人麻雀の手を選択し放銃することがある．これを解決するためには一人麻雀の評価値と予測した危険度をもとに学習を行い，手を決定するモデルを構築する必要がある．

降りが上手くなったため，役の無い鳴きや牌効率の悪い手といった一人麻雀の手の粗さも目立つ．相手の聴牌率や降りのデータに一人麻雀の手を使用しているため，一人麻雀の手の改善は全体の改善につながる．また今は面前で聴牌したらすべてリーチやリーチ後の可能な牌はすべて暗槓といったヒューリスティックな部分も残っており，より精度の高い打牌をするためにリーチを打つか打たないかといった学習も行わなければならない．これらの解決のため一人麻雀の特徴量やモデルについも見直す必要がある．

9. おわりに

本研究では相手の抽象化したモデル化を考え，牌譜をもとに学習を行い，その予測結果とモンテカルロ法を用いることでコンピュータ麻雀プレイヤの手を決定するアルゴリズムを提案した．各抽象化した要素の予測は人間と比較しても精度の高く，高速に行えた．実際の手を決定する局面においても今までの行われていた，降りるべき局面のタグを一切使うことなく降りと攻めのバランスを人間並みに行うことが可能になった．結果として天鳳 [11] において 2,227 試合，戦わせた結果，安定レーティングが 1681 点という上級卓の平均並みの実力を得た．

今後の課題としては現在の得点状況からに応じて手を選択するモデルの構築である．今のモデルは現在の点数状況が入っておらず，どのような状況であっても同じ手を指す．当然，上級者は今の点数状況に応じて，何点の和了をすべきか考えており，またリーチなどにどれだけ攻めるか降りるかを判定している．これらの判断は最終的な順位に大きく影響するため実力向上に必須である．これらの判断にはシミュレーション中の利益をゲームの得点では無く，順位に影響する何らかの値にする必要がある．例えば現在の点数状況からゲーム終了時の順位分布を予測が可能になれば，期待順位を計算しシミュレーションの利益とすることが可能になる．

参考文献

[1] Bowling, M., Risk, N. A., Bard, N., Billings, D., Burch, N., Davidson, J., Hawkin, J., Holte, R., Johanson, M., Kan, M. et al.: A demonstration of the Polaris poker system, *Proceedings of 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pp. 1391–1392 (2009).

[2] Van Der Kleij, A.: Monte Carlo Tree Search and Opponent Modeling through Player Clustering in no-limit Texas Hold' em Poker, Master's thesis, University of Groningen (2010).

[3] Davidson, A.: Opponent modeling and search in poker: Learning and acting in a hostile and un-certain environment, Master's thesis, University of Alberta (2002).

[4] Buro, M., Long, J. R., Furtak, T. and Sturtevant, N. R.: Improving State Evaluation, Inference, and Search in Trick-Based Card Games., *Proceedings of 21th International Joint Conference on Artificial Intelligence*, pp. 1407–1413 (2009).

[5] 水上直紀，中張遼太郎，浦　晃，三輪　誠，鶴岡慶雅，近山　隆：多人数性を分割した教師付き学習による四人麻雀プログラムの実現，情報処理学会論文誌 (in press).

[6] 我妻　敦，原田将旗，森田　一，古宮嘉那子，小谷善行：SVR を用いた麻雀における捨て牌の危険度の推定，情報処理学会研究報告. GI,[ゲーム情報学]，Vol. 2014, No. 12, pp. 1–3 (2014).

[7] 三木理斗，近山　隆：多人数不完全情報ゲームにおける最適行動決定に関する研究，修士論文，東京大学 (2010).

[8] Coulom, R.: Efficient selectivity and backup operators in Monte-Carlo tree search, *In Proceedings of the 5th International Conference on Computer and Games*, Springer, pp. 72–83 (2006).

[9] Duchi, J. and Singer, Y.: Efficient online and batch learning using forward backward splitting, *The Journal of Machine Learning Research*, Vol. 10, pp. 2899–2934 (2009).

[10] Duchi, J., Hazan, E. and Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization, *The Journal of Machine Learning Research*, Vol. 12, pp. 2121–2159 (2011).

[11] 角田真吾：天鳳，`http://tenhou.net/` (2014).

[12] kmo2: まったり麻雀，`http://homepage2.nifty.com/kmo2/` (2014).

[13] とつげき東北：保障安定レーティング，`http://totutohoku.b23.coreserver.jp/hp/SLtotu14.htm` (2001).