

麻雀における他家の待ち牌推測

松田, 真治
九州大学大学院システム情報科学府

伊東, 栄典
九州大学情報基盤研究開発センター

<https://hdl.handle.net/2324/4740676>

出版情報：情報処理学会研究報告. 2020-MPS-130 (4), pp.1-6, 2020-09-22. 情報処理学会
バージョン：

権利関係：本論文は著作権者である情報処理学会の許可のもとに掲載しています。ご利用の際は著作権法に従ってください。

麻雀における他家の待ち牌推測

松田真治¹ 伊東栄典²

概要：本研究の目的は、NNを用いて麻雀における他家の待ち牌の予測をすることである。先行研究において、相手の状態を推測する場合、時系列のない情報を利用していた。そこで本研究では、牌の切られた順序を入力として学習を行った。その結果、先行研究と比較して、4.16% f1 score を向上させることができた。

キーワード：麻雀, ニューラルネットワーク, 待ち牌推測, 牌譜

Estimating the waiting tiles of other players in Mahjong

MATSUDA SHINJI¹ EISUKE ITO²

Abstract: The purpose of this study is to use NN to predict the waiting tiles of other players in Mahjong. In previous studies, information without a time series was used to estimate the opponent's state. In this study, the order in which the tiles were discarded was used as input for learning. As a result, we were able to improve the f1 score by 4.16% compared to the previous study.

Keywords: Mahjong, Neural network, Waiting tiles estimation, Mahjong game record

1. はじめに

麻雀とは不完全情報ゲームである。将棋や囲碁のようにすべての情報がプレイヤーに開示されている完全情報ゲームと異なり、限られた情報の中から他のプレイヤーの状況を推察しつつ自分に与えられた手の価値の期待値を最大化するような選択を行い続けるゲームである。

例えば将来的には高くなりそうな手ももらう場合でも、他のプレイヤーにスピード感を感じた場合には、自分の手の価値を下げてでも速くあがることを優先したり、自分の手が安く勝負をする価値が低い場合でも、のびのびとしたプレーをさせないように、ときにはブラフを用いてでも他のプレイヤーの足止めをしたりもする。このように麻雀のような不完全情報ゲームでは自分の都合のみでプレイの選択を行うことは少なく、常に他のプレイヤーの状況を推察しながら臨機応変に立ち回りをすることが重要である。

周りの状況に臨機応変に対応しながら自分の意思決定を行う様は現実世界でいうところの押し引きと非常に類似している。したがって麻雀のような不完全情報ゲームに関する研究を行うことによって、現実で起こる押し引きの絡むような問題を解決する手助けとなる可能性があり、研究の意義は十分にあると言える。

本研究の目的はオンライン麻雀サイト天鳳の牌譜を教師データとして学習を行ってモデルを作成し、そのモデルを用いて、押し引きの際に特に重要な要素の一つである他のプレイヤーの待ち牌の予測の性能を評価することである。

先行研究[1][2][3][4][5]で、他のプレイヤーの捨て牌から待ち牌の推測を行う際に用いられる情報は、各色の捨てられた枚数や捨てられた牌の種類の数、手出しの回数、捨てた中張牌の種類数などの情報である。これらの情報は牌が捨てられた順序は考慮されていない。

しかし一般的には、捨て牌から他のプレイヤーの待ち牌の予測を行う際に、相手がどの順序で牌を切ったかという情報は極めて重要なものだと考えられている。そこで本研究では待ち牌予測に捨てられた牌の順序を考慮する手法を提案し、その予測性能を評価した。

2. 捨て牌の打牌順序と待ち牌の推測

2.1 捨て牌から他家の待ち牌を推測する際の打牌順序の重要性

捨て牌から他家の待ち牌を読む場合に、打牌順序の情報が重要になる理由について例を用いて説明する。

まず麻雀において、あるプレイヤーが五萬→七萬の順番で牌を捨てている場合に、そのプレイヤーに対して三萬と六萬が比較的に安全だということが知られている。しかしながら、あるプレイヤーが七萬→五萬の順序で牌を捨てている場合は、三萬と六萬が安全とは言えない。

このように、Aの牌を捨てた後にBの牌を捨てた、という情報を材料に、他のプレイヤーの待ち牌予測を行う機会は麻雀において頻出する。

1 九州大学 システム情報科学府
Faculty of Information Science and Electrical Engineering, Kyushu University
2 九州大学情報基盤研究開発センター
Research Institute for Information Technology, Kyushu University

単に A を捨てた、B を捨てたという情報だけではこのような予測にたどり着くことはできない。

そこで本研究では、捨て牌のデータを、単なる捨て牌の集合で表すのではなく、打牌された順序を織り込みつつ表現する手法を考えた。

2.2 手出しとツモ切り

麻雀では牌の切り方に 2 つのパターンがある。手出しとツモ切りである。手出しとは牌を山からツモした後、もともと手牌にあった牌を切ることである。それに対してツモ切りとは牌を山からツモした後、そのツモした牌をそのまま切ることである。この 2 つは似ているようで大きく意味が異なる。

手出しされた牌とは、何らかの理由で手牌に残していた牌に比べ、その後ツモしてきた牌の価値がより高かったために切られた牌である。言い換えれば捨て牌にある牌よりは価値が高く、手牌にある牌よりは価値が低い牌である。

一方ツモ切りされた牌とは手牌にある牌よりは価値が低い、捨て牌にある牌より価値が高いとは必ずしも言えない。

このように手出しとツモ切りでは捨て牌の持つ情報が異なるため、たとえ同じ一萬だとしても手出し一萬とツモ切り一萬では違う牌とみなしたほうが機械学習においては有効である。

3. 麻雀サイト天鳳の牌譜データ

機械学習による待ち牌推測の性能評価のために、本研究ではオンライン麻雀サイト天鳳の牌譜データを利用した。

天鳳とは最大手のオンライン麻雀サイトであり、オンラインで麻雀対戦できる一方、その対戦データは牌譜として記録され、天鳳の開発者によって無償で公開[6]されている。

本研究では天鳳五段以上の実力者のみがプレイできる鳳凰卓の牌譜を性能評価のために利用した。期間は 2017 年度のもので、ルールは最も一般的なルールである四人打ち、東南戦、赤有り、喰断么九有りのものに限定した。

この条件を満たす XML ファイルの件数は 199949 である。また、総和了件数は 1670121、10 回以上打牌している総和了件数は 1092518 である。

天鳳の牌譜は XML ファイルによって特殊な形式で表現されているため、研究で用いるには解析を行う必要がある。解析については小林聡氏のブログ[7][8][9][10]を参考にした。解析した牌譜から得られる情報を機械学習の入力で扱うためのベクトル化手法は、次章で説明する。

牌譜の解析について説明する。

天鳳の牌譜は拡張子こそ XML であるが、XML の仕様に準拠しておらず、天鳳の独自構文である。

天鳳の牌譜は 1 つの XML ファイルが 1 半荘を表している。ゲーム中に起こる事象はタグとその属性で表現され、事象が起きた順に並べられている。閉じタグは存在しない。

<INIT>タグは局の開始を表現し、<AGARI>タグ、<RYUKYOKU>タグはそれぞれ、和了があった場合の情報、和了がなかった場合の情報が表現されている。<AGARI>タグも<RYUKYOKU>タグも共に局の終了を表現するタグでもある。すなわち<INIT>タグから<AGARI>タグまたは<RYUKYOKU>タグまでで一局を表現している。1 つの XML ファイルにこの<INIT>タグから<AGARI>タグまたは<RYUKYOKU>タグまでの集合が局数分書かれている。ただしダブロンが生じた場合は<INIT>(途中の情報)<AGARI><AGARI>などの形で表現される。

開局から終局までのツモ情報や打牌情報などは<INIT>タグ~<AGARI>タグまたは<RYUKYOKU>タグの間に羅列されている。本研究では和了の情報のみを利用するため、<INIT>タグ~<AGARI>タグまでの情報のみを利用した。

本研究で利用したタグについて簡潔に説明する。まず前提として、天鳳の牌譜では 136 枚の牌に 0~135 までの牌番号が振られている。

ツモ情報は<T(牌番号)>、<U(牌番号)>、<V(牌番号)>、<W(牌番号)>の形で表記されている。T,U,V,W は誰がツモしたかを表現している。

打牌情報は<D(牌番号)>、<E(牌番号)>、<F(牌番号)>、<G(牌番号)>の形で表記されている。D,E,F,G は誰が打牌したのかを表現している。

T, U, V, W の表すプレイヤーはそれぞれ D, E, F, G の表すプレイヤーと対応している。

例えば A さんが 0 の牌をツモって 0 の牌を打牌し、その次に B さんが 1 の牌をツモして 2 の牌を打牌した場合は<T0><D0><U1><E2>のように表現される。打牌が手出しだったかツモ切りかなどの情報は牌譜には含まれていないのでツモした牌の牌番号と捨てた牌の牌番号が同値かどうかを比較して、自分で判定を行う必要がある。

和了情報は<AGARI>タグで表現されている。AGARI タグは複数の属性を持つ。本研究で利用したのは和了したプレイヤーが誰かを表す who 属性、和了したときの手牌を表す tehai 属性、どの牌で和了したかを表す machi 属性である。

machi 属性はどの牌で和了したかを表現しており、厳密には待ち牌を表現しているわけではない。例えば下の図 3-1 の和了例を見てみる。





図 3-1. 和了の例

上の例について9筒で和了したとする。すると牌譜上のAGARI タグの machi 属性には9筒に対応する牌番号が記録される。しかし、実際は6筒も待ち牌である。したがって厳密な待ち牌は牌譜に記録されていないため、自分で検出しなければならない。その手順について説明する。

まず tehai 属性から手牌情報を抜き出す。次に machi 属性からどの牌で和了したかを抜き出し、その牌を手牌から消し、和了する一步手前の状態を再現する。次に34枚の牌全てについて、1萬 + 手牌, 2萬 + 手牌, ..., 西 + 手牌, 北 + 手牌 が和了しているかどうかを判定する。和了状態ならば手牌に足したその牌は待ち牌である。和了であるかは深さ優先探索などで判定する。

4. 牌譜データのベクトル化

4.1 打牌にツモ切り・手出しの情報を持たせる表現

ツモ切りした牌と手出した牌を違う牌だとみなすと、牌の種類は合計74種類になる。そこで表4-1に示す対応で、0~73の番号でツモ切り・手出しの情報を持つ牌を表現できる。

表 4-1. 牌と番号の対応付け

牌	番号
ツモ切り 1 萬~9 萬	0~8
ツモ切り 1 筒~9 筒	9~17
ツモ切り 1 索~9 索	18~26
ツモ切り 東、南、西、北	27~30
ツモ切り 白、發、中	31~33
ツモ切り 赤 5 萬、赤 5 筒、赤 5 索	34~36
手出し 1 萬~9 萬	37~45
手出し 1 筒~9 筒	46~54
手出し 1 索~9 索	55~63
手出し 東、南、西、北	64~67
手出し 白、發、中	68~70
手出し 赤 5 萬、赤 5 筒、赤 5 索	71~73

4.2 打牌順序を考慮しない捨て牌のベクトル化

打牌順序を考慮しない捨て牌とは、要するに何を切ったかという情報である。図4-1の捨て牌からのデータ抽出を考える。

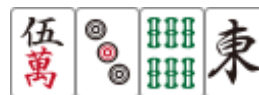


図 4-1. 捨て牌例 (すべてツモ切りとする)

機械学習の入力ベクトルとして、初期値0の74bitの数列を考える。次に捨て牌のそれぞれの牌番号について対応するbitを1にする。図4-1の場合は4,11,23,27bit目の値を1にする。

0bit	1bit	~	4 bit	~	11 bit	~	73 bit
0	0	~	1	~	1	~	0

図 4-2. 打牌順序を考慮しない捨て牌を表すベクトルのイメージ

4.3 打牌順序を考慮する捨て牌のベクトル化

順序関係の情報を持つ捨て牌のベクトル化について説明する。

その方法は、任意の2枚の牌を捨てる順序対について一意に番号を割り当てて表現する方法である。2枚の牌の組み合わせは高々 $74 * 74 = 5476$ である。予め2枚の牌の打牌順序を0~5475の番号に対応する方法を決めることで、任意の打牌の順序関係を表現できる。

本研究では(先に捨てた牌の牌番号) * 74 + (後に捨てた牌の牌番号)の式を用いて、打牌順序を保つ2枚の牌の組み合わせを番号で表現する。

例として図4-1の捨て牌から、打牌順序を考慮する捨て牌のベクトル化を示す。

はじめに捨て牌からすべての打牌順序を抜き出す。図4-1の場合は(ツモ切り5萬, ツモ切り3筒), (ツモ切り5萬, ツモ切り6索), (ツモ切り5萬, ツモ切り東), (ツモ切り3筒, ツモ切り6索), (ツモ切り3筒, ツモ切り東), (ツモ切り6索, ツモ切り東)である。これを牌番号に置き換える。置き換えたものが(4,11), (4,23), (4,27), (11,23), (11,27), (23,27)である。これらの番号の組み合わせを上記の(先に捨てた牌の牌番号) * 74 + (後に捨てた牌の牌番号)という式にあてはめるとそれぞれ307,319,323,837,841,1729となる。

よって入力ベクトルとして初期値0の5476bitの数列を考える。次に上で求めた番号について対応するbitを1にする。図4-1の場合は307, 319, 323, 837, 841, 1729 bit目の値を1にすればよい。

0 bit	~	307 bit	~	1729 bit	~	5475 bit
0	~	1	~	1729	~	0

図 4-3. 打牌順序を考慮する捨て牌を表すベクトルのイメージ

以上の方法で、任意の2つの牌を切った順序をベクトル化できる。また、入力に打牌順序の情報を与えることができる。

5. 実験手法

5.1 実験設定

和了する一手手前の状態を牌譜から抜き出し、以下の2通りの方法でベクトル化する。

- ・打牌順序を考慮しない捨て牌
- ・打牌順序を考慮する捨て牌

また、N回打牌を行った捨て牌について、打牌順序を考慮しない捨て牌は高々N個の牌の集合であるのに対し、打牌順序を考慮する捨て牌は高々 $\binom{n}{2}$ 個の打牌の順序対の集合である。したがって打牌回数が増えれば増えるほど、打牌順序を考慮する場合の捨て牌の情報量が、打牌順序を考慮しない捨て牌の情報よりも大きくなると推測し、以下の2通りの方法でもベクトル化した。

- ・打牌順序を考慮しない捨て牌（打牌10回以上）
- ・打牌順序を考慮する捨て牌（打牌10回以上）

以上の4つのベクトル化された情報を入力として機械学習を行い、待ち牌の推測性能について評価を行う。以降、簡単のため各ベクトル化を入力として学習したモデルを以下のように呼ぶ。

5.2 アンダーサンプリング

本研究では、ある牌が待ち牌となるサンプルを正例、待ち牌でないサンプルを負例とする。

ある牌に着目したとき、全サンプルのうちの正例となるサンプルは1%~10%程度である。このデータをそのまま用いて機械学習すると、ほとんどの場合において待ち牌ではないと推測する学習器ができ、accuracy90%以上を達成することができる。しかしでは意味がない。

そこで学習を行う前に、データの負例の数を正例の数に合わせる事により、意味のある結果を出力する学習器を作成できる。この作業はアンダーサンプリングと呼ばれる。

本研究では数牌については正例40000件、負例40000件になるようにサンプリングをした。字牌については正例が少ないため、正例10000件、負例10000件になるようにサンプリングをした。

またテストで用いるサンプルはアンダーサンプリングをしていないデータからランダムに8万件を選んだ。

5.3 実装

Chainerで3層ニューラルネットを構築して学習をした。第1層の入力数は入力次元数、出力数は100。第2層の入

力数は100、出力数は100。第3層の入力数は100、出力数は2。最適化関数はAdamを利用した。エポック数50、バッチサイズ16で学習をした。データは70%を訓練データセットとして、30%を検証データセットとして利用した。

6. 実験結果と考察

6.1 捨て牌の打牌順序を考慮しない場合と考慮する場合の比較

(a)打牌順序を考慮しない捨て牌

(b)打牌順序を考慮する捨て牌

を入力として、それぞれ学習をした。それぞれの学習器のf1scoreを比較した結果を図6-1に示す。

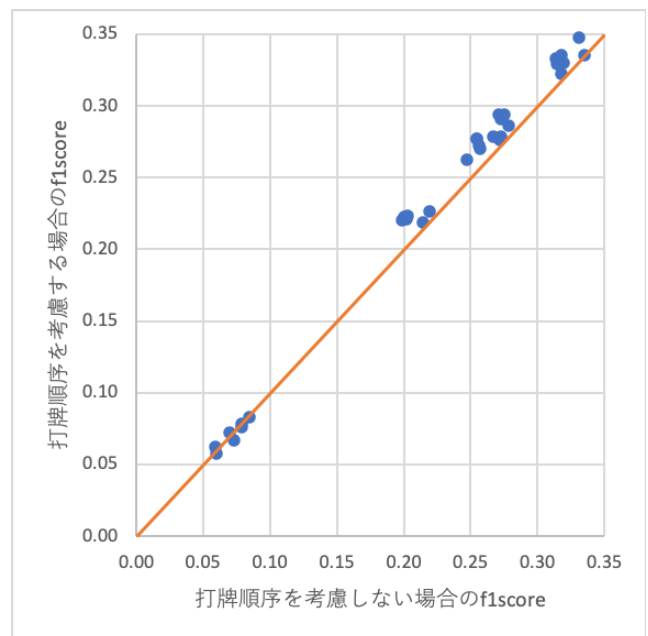


図6-1. (a)と(b)の比較

図6-1に示すように、打牌順序を考慮しない場合と打牌順序を考慮する場合を比較すると、打牌順序を考慮したモデルのほうが平均して4.16%高いf1scoreを記録した。ただし、南、北、白、発、中の5種類の牌においては打牌順序を考慮しない場合の方がf1scoreが高かった。

6.2 打牌回数を10回以上に制限した場合のf1score増加率の比較

(c)打牌順序を考慮しない捨て牌（打牌10回以上）

(d)打牌順序を考慮する捨て牌（打牌10回以上）

を入力として、それぞれ学習をした。

(a)と(c)のf1scoreの比較を図6-2に、(b)と(d)のf1scoreの比較を図6-3に示す。

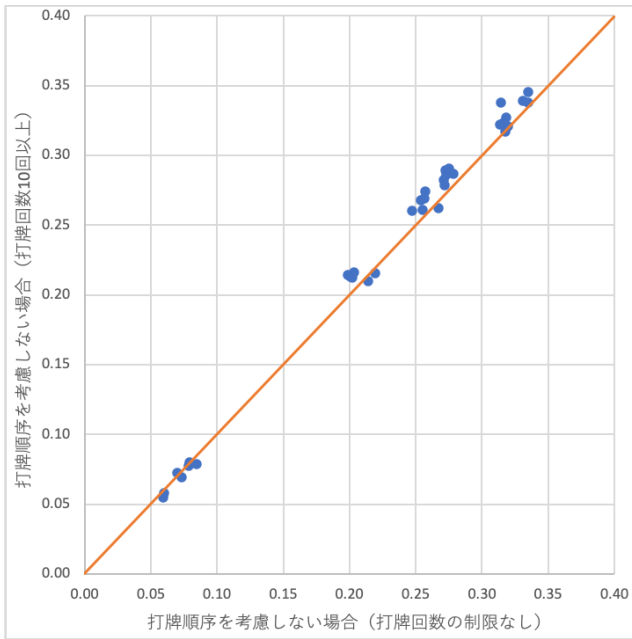


図 6-2. (a) と(c)の比較

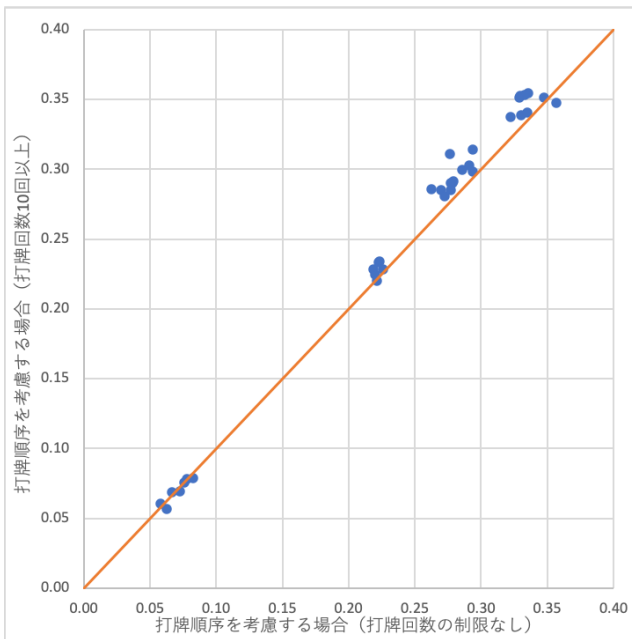


図 6-3. (b)と(d)の比較

図 6-2 に示すように、打牌順序を考慮しない場合、打牌回数を 10 回以上に制限すると、平均で 2.12% f1score が高くなった。

図 6-3 に示すように、打牌順序を考慮する場合、打牌回数を 10 回以上に制限すると、平均で 2.96% f1score が向上した。

6.3 考察

打牌の順序を考慮する捨て牌を入力として機械学習を行えば、考慮しない場合を入力として学習した場合と比較して飛躍的に良い待ち牌予測の精度が出ると想定していた。しかしながら、実際は平均で見ると 4.16% しか f1score が増加していない。

この原因としては、捨て牌の打牌のすべての順序関係を捨ってしまうことにより、ノイズとなるような情報も拾われたことが考えられる。例えば第 1 打と第 16 打の打牌の関係に重要な情報が含まれることは殆どないが、本研究の手法では第 1 打と第 16 打との打牌順序にも意味を見出そうとする。

また、打牌順序を考慮することで予測精度が下がる場合もある。(a)の場合、南、北、白、発、中の予測精度が下がり、(c)の場合は東、南、西、白、発の予測精度が下がった。

予測精度が下がった牌は全て字牌である。これは直感的には理解のできる結果である。字牌は数牌と違い順子を作ることが出来ず、打牌の順序に意味を持たないことが多いためである。

また、(b)から(d)の f1score の上昇幅は(a)から(c)の f1score の上昇幅より大きくなると予想していたが、牌によっては予測の精度が下がるものもあった。この原因は打牌の回数を 10 回以上に制限することで、前述したノイズとなる情報を拾いやすくなるためだと考えられる。

7. まとめと今後の課題

本研究では麻雀の和了局面における待ち牌予測を行った。オンライン麻雀ゲーム天鳳の牌譜データを用いて、機械学習で待ち牌を予測した。

本研究では新たな手法として、捨て牌の順序を考慮した手法を提案した。

学習に用いる捨て牌情報について、打牌の順序を考慮することで、打牌の順序を考慮しない場合と比較して、待ち牌の予測の f1score が平均 4.16% が向上した。

打牌の順序関係を考慮しない場合、打牌の回数を 10 回以上に制限すると、打牌の回数を制限していないときと比較して、待ち牌予測の f1score が 2.12% 向上した。

打牌の順序関係を考慮する場合、打牌の回数を 10 回以上に制限すると、打牌の回数を制限していないときと比較して、待ち牌予測の f1score が 2.96% 向上した。

したがって打牌回数を制限したことによる f1score の上昇幅は、打牌順序を考慮する場合の方が考慮しない場合と比較して大きかった。

今後の課題としては、予測精度のさらなる向上のために、打牌の順序対に重み付けを行うことが考えられる。

例えば、打牌された間隔が長い順序対は、情報の重みを小さくし、逆に間隔が短い場合は重みを大きくする、序盤

の打牌の情報より終盤の打牌の情報の重みを大きくする、ドラや安牌などの情報を考慮に入れる、などの工夫が考えられる。

参考文献

- [1] 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危険度の推定. 報告ゲーム情報学(GI), Vol. 2014-GI-31, No. 12, pp. 1-3, 2014-03-10
- [2] 栗田萌,保木邦仁. 麻雀における他家の手牌と待ちの予測に基づく放銃確率推定. 報告ゲーム情報学(GI), Vol. 2017-GI-38, No. 5, pp. 1-8, 2017-07-08
- [3] 矢ノ口裕貴,篠埜功. ニューラルネットワークを用いた麻雀の捨て牌危険度推定. 第 79 回全国大会講演論文集, Vol. 2017, No. 1, pp. 469-470, 2017-03-16
- [4] 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 4 人麻雀プログラムの実現. 情報処理学会論文誌, Vol. 55, No. 11, pp. 2410-2420, 2014-11-15
- [5] 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. ゲームプログラミングワークショップ 2007 論文集, Vol. 2007, No. 12, pp. 76-83, 2007-11-09
- [6] オンライン対戦麻雀 天鳳 / ログ,
<https://tenhou.net/sc/raw/>
- [7] 天鳳の牌譜を解析する(1),
<https://blog.kobalab.net/entry/20170225/1488036549>
- [8] 天鳳の牌譜を解析する(2),
<https://blog.kobalab.net/entry/20170228/1488294993>
- [9] 天鳳の牌譜を解析する(3),
<https://blog.kobalab.net/entry/20170312/1489315432>
- [10] 天鳳の牌譜を解析する(4),
<https://blog.kobalab.net/entry/20170720/1500479235>