

将棋プレイヤーモデルによるプレイヤーの棋力識別

落合竜一 佐藤泰介

東京工業大学情報理工学研究科計算工学専攻

概要

将棋は2人零和有限確定完全情報ゲームであり、理論的には最善手順が存在するので、将棋に勝つためのプレイヤーモデリングでは確率的な要素を必要としない。しかし、実際に対局するプレイヤーは、焦り、迷い等、精神状態の影響を受けるので、ある局面に対して常に同じ行動をとるわけではない。従って、対局は不確実性を含む現象であり、モデル化によって対局現象を分析することができる。本稿では、記述言語としてPRISMを用い、プレイヤーが着手を選択する過程のモデル化によって将棋プレイヤーモデルを構築する。また、将棋棋譜データベース将棋倶楽部24万局集 [1] を用い、本モデルの確率パラメータ学習実験を行う。まず、付属するレーティング情報に基づいて棋力ごとに分類し、各棋力についてこれらの棋譜の一部を訓練データとし、PRISM 付属のEM アルゴリズムを用いて各棋力毎の確率パラメータ推定実験を行う。また、推定したパラメータの妥当性を検査するために、各棋力について訓練データとは異なる棋譜を同様に将棋倶楽部24万局集から抽出し、これをテストデータとしてプレイヤーの棋力の識別実験を試みる。

Identifying shogi strength by a Shogi Player Model

Ryuuichi Ochiai Taisuke Sato

Dept. of Computer Science, Tokyo Institute of Technology

Abstract

In this paper, we take the shogi game as an instance of probabilistic events, write a shogi player model in PRISM. The shogi game is a complete information game and hence there is a winning strategy so that there is no need for probabilistic modeling to win the game. But as a real shogi player does not always make the same choice at each position because of the mental state of him or her, the shogi game is considered as a probabilistic event, and this enables us to analyze the game by the model on the basis of the generation process of the game. The shogi player model constructed in this paper models a decision process of a player's choice. We also perform an experiment of learning statistical parameters with a shogi database. First, the game records are divided into classes according to the player's shogi strength based on their rating. Then, we take training data from each class, and learn the statistical parameters for each class. Next we take test data from each class, and perform an experiment of identification of the player's shogi strength. We experimentally show an effectiveness of our model.

1. はじめに

現在、自然現象や社会現象等、発生過程に不確実性を含む現象について研究を行う際、隠れマルコフモデル、ベイジアンネットワーク等の統計モデルがよく用いられている。統計モデルを適用すると、現象の発生過程を考慮しながらモデル化を行い、現象の実データから最尤推定法によってモデルの確率パラメータを推定し、モデル及び推定されたパラメータを用いて現象の分析を行うことができる。本稿では、将棋の対局現象におけるプレイヤーの不確実性に

注目し、これを分析するために将棋プレイヤーモデルを構築する。

将棋は2人零和有限確定完全情報ゲームであり、理論的には最善手順が存在するので、将棋に勝つためのプレイヤーモデリングでは確率的な要素を必要としない。しかし、実際の対局を行うプレイヤーは全ての手を読みきれるわけではなく、幾つかの手の候補から一つを選択しなければならない。また、その際焦り、迷い等、精神状態の影響を受けるので、ある局面に対して常に同じ選択をするわけではない。従

って、対局は不確実性を含む現象であると言える。

このような不確実性を含む現象のモデリングを行う場合、隠れマルコフモデルやベイジアンネットワークがよく用いられる。

しかし、これらの統計モデルを用いてモデリングを行う場合、将棋の全盤面を考慮する必要がある。将棋のパターン数は膨大なので、実際にこれらを用いて将棋プレイヤーのモデルを構築することは困難である。

また、将棋のような複雑な現象をモデル化する場合、完成したモデルも複雑になることが予想される。複雑なモデルのパラメータを学習する場合、データスパースネスの問題を考慮する必要がある。しかし、近年インターネット将棋道場等、インターネットを用いた将棋の対局が普及しており、その棋譜は将棋倶楽部24万局集 [1] のような形で大量に入手可能である。従って、将来的にはデータスパースネスの問題を解消できる可能性がある。

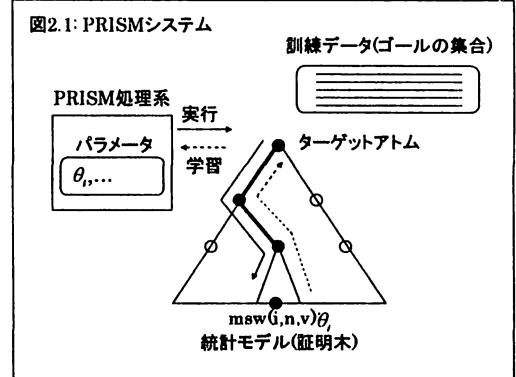
本稿では、記述言語として記号的統計モデリング言語 PRISM (PRogramming In Statistical Modeling) [2] を用い、将棋プレイヤーモデルの構築を行う。PRISM 言語は汎用の統計モデル記述言語で、その処理系は不完全データからの最尤推定である EM (Expectation-Maximization) アルゴリズムによるパラメータ学習機能を備えている。そして、将棋の全盤面は初期状態と規則の複数回適用から得られるので、PRISM でサポートされている再帰を用いれば将棋プレイヤーモデルを実際に構築することができる。また、本モデルを評価するために、将棋倶楽部24万局集 [1] を用いてモデルのパラメータ学習実験と将棋プレイヤーの棋力識別実験を行う。

本稿の構成は次のようになっている。まず、2節ではまず PRISM について説明を行う。そして、3節で構築する将棋プレイヤーモデルの詳細を説明し、4節でプレイヤーモデルのパラメータ学習実験と棋力識別実験を説明する。最後に5節でまとめと今後の課題を述べる。

2. PRISM

本節では、PRISM 言語とその処理系について説明する。

まず、図 2.1 に PRISM システムの全体像を示す。



ユーザは、まず実世界の観測事象に対応するターゲットアトムを考える。次に、このターゲットアトムを説明する統計モデルを記述する。これが PRISM プログラムである。PRISM プログラムは Prolog 節の形式で記述できる。ただし、節の定義中に確率的スイッチ ($ms_w(i, n, v)$) を呼び出す。確率的スイッチとは、ターゲットアトムの証明木の葉ノードに対応する。確率的スイッチは、各々が真になる確率 (パラメータ) θ を持っている。

一方、実世界で観測されたアトムを観測アトムまたはゴールと呼び、独立に観測された観測アトムの集合を訓練データと呼ぶ。統計モデルの学習では訓練データが与えられたときターゲットアトムの確率が観測アトムの統計分布に近づくように確率的スイッチのパラメータを調節する。また、実行では確率的スイッチのパラメータに従ってターゲットアトムを確率的に生成する。

2.1. 確率的スイッチ

確率的スイッチ $ms_w(i, n, v)$ が表現する確率分布の条件は次のとおりである。

1. PRISM プログラム中の $ms_w(i, n, v)$ は各 i に対し、多値スイッチ宣言により事前に基底項の集合 V_i が与えられており、 $v \in V_i$ である。 i はスイッチ名を表す。
2. 任意の i と $V_i = \{v_1, \dots, v_k\}$ に対して、任意の n について $ms_w(i, n, v_1), \dots, ms_w(i, n, v_k)$ のいずれか 1 つのみが真となり、残りは全て偽になる。また、 $v \in V_i$ について $ms_w(i, n, v)$ が真になる確率をパラメータ θ_i を使って $\theta_i(v)$ と表す。すると $\sum_{v \in V_i} \theta_i(v) = 1$ である。
3. $n \neq m$ ならば $ms_w(i, n, *)$ と $ms_w(i, m, *)$ は互いに独立であり、かつ同一分布 θ_i に従う。
4. $i \neq j$ ならば、 $ms_w(i, *, *)$ と $ms_w(j, *, *)$ の真偽は互いに独立である。

確率的スイッチは統計モデル中の節の体部ならば

どこにでも記述することができるが、所望の確率分布を統計モデルとして厳密に実現するためには上記の条件を満たす必要がある。

あるアトムが確率的スイッチを呼び出すならばそのアトムは真偽が確率的に定まるアトムであり確率のアトムと呼ぶ。確率のアトムの述語を確率述語と呼ぶ。確率的でないアトムを非確率アトムと呼び、非確率のアトムの述語を非確率的述語と呼ぶ。PRISM プログラム中の統計モデルは確率的述語及び確率述語から呼び出される非確率的述語と定めることができる。

2.2 PRISM プログラムの例

PRISM プログラムの例として次のような問題を考える。

コインを5回投げて、表と裏のどちらが上になったかを観測する。コインには通常のコインといかさまコインが存在する。通常のコインは表が上になる確率と裏が上になる確率が等しいが、いかさまコインは裏のみが上になる。観測者はこれらのコインを確率的に選択してコインを投げることにする。

コインの表裏の確率分布を図 2.2 のように表現する。ターゲットアトムは `coin/5` であり、1 行目のターゲット宣言で指定されている。2 行目にはデータファイル宣言があり、訓練データが `coin.dat` であることを指定している。3 行目から 5 行目には多値スイッチ宣言があり、各スイッチの基底項集合を定めている。6 行目以降が統計モデルである。そして、図 2.3 のような訓練データを与え PRISM 内蔵の EM 学習コマンドを実行すると、図 2.3 のような確率パラメータが得られ、この人はほぼ二回に一回いかさまコインを利用していたことが分かる。

図2.2 コインの表裏モデル

```
01: target(coin,1).
02: data('coin.dat').
   %普通のコイン
03: values(coin,[0,1]).
   %いかさまコイン
04: values(coin0,[0]).
   %コインの選択
05 values(select_coin,[coin,coin0]).
06: coin([X1,X2,X3,X4,X5]):-
   dice(1,Xs).
07: coin(_,_).
08: dice(N,[X|Xs]):-
   %コインを選択
   msw(select_coin,N,Coin),
   %コインを投げる
   msw(Coin,N,X),
   N1 is N + 1,
   dice(N1,Xs).
```

図2.3 訓練データと学習結果

```
訓練データ
coin([0,1,0,0,0]).
coin([0,1,0,0,1]).
coin([1,0,1,0,0]).
coin([0,1,1,0,0]).
coin([0,1,1,0,1]).

学習した確率パラメータ
Switch coin0: 0 (1.0)
Switch coin:  0 (0.250...) 1 (0.749...)
Switch select_coin: coin (0.533...) coin0 (0.466...)
```

3. 将棋プレイヤーモデルの概要

本節では、構築する将棋プレイヤーモデルの詳細を説明する。

3.1. 用語の定義

まず、本モデルで用いる用語を定義する。

- 対局：将棋の試合。
- プレイヤ：対局を行う人物。
- 盤面：将棋盤上に乗っている駒の配置と各駒の向きや表裏を組み合わせた状態。
- 持ち駒：各対局者が盤上以外に保持している駒。
- 手番：次に手を指すプレイヤーを表す状態。
- 投了：手番側のプレイヤーが負けを認める行為。
- 先手：最初の手番のプレイヤー。
- 後手：先手ではないプレイヤー
- 局面：盤面、持ち駒、および手番を組み合わせた状態。
- 手：ある局面から次の局面への遷移。
- 着手：対局時に決定する手。
- 指す：着手を行うこと。
- 打つ：持ち駒を用いた手を指すこと。
- 成る：盤上にある駒を移動して裏返す手を指すこと。
- 手数：現局面までに両プレイヤーが指した手の数。
- 棋譜：対局の記録。両プレイヤーの名前と着手の列で構成される。
- 棋力、レーティング：プレイヤーの強さ。

3.2. モデル化の対象

本モデルで用いる訓練データは棋譜である。従って、PRISM を用いてモデルのパラメータ学習を行うためには、棋譜を生成する現象を PRISM プログラムで記述する必要がある。しかし、本モデルは将棋プレイヤーのモデルである。よって、棋譜生成モデルが将棋プレイヤーのモデルによって決まることを示す必要がある。

3.2.1 記号の定義

まず、ここで用いる記号を定義する。

- t_1 : 先手プレイヤー
- t_2 : 後手プレイヤー
- n : 手数
- m_1, \dots, m_n : 第1手目から第n手目までの着手
- k_0 : 初期局面
- k_1, \dots, k_n : 1手の着手後の局面から最終局面までの各局面

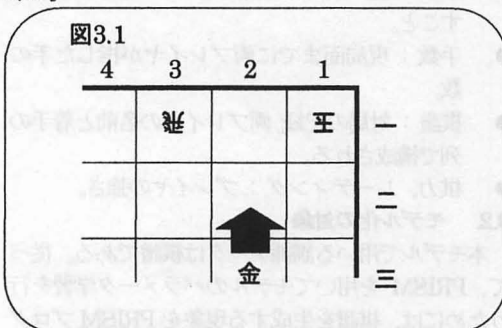
また、各記号を大文字にしたものを用いてそれぞれの確率変数を表すことにする。

3.2.2 数式による定義

ある棋譜が生成される現象の同時確率分布は $P(M_1, \dots, M_N, T_1, T_2)$ と表される。本モデルでは両プレイヤーを固定したときにある棋譜が生成される現象をモデル化するので、本モデルが表現すべき確率分布は $P(M_1, \dots, M_N | T_1, T_2)$ となる。ここで、条件付確率の定義を繰り返し適用すると、次のようになる

$$P(M_1, \dots, M_N | T_1, T_2) = \prod_{i=1}^N P(M_i | M_1, \dots, M_{i-1}, T_1, T_2)$$

ここで、 $P(M_i | M_1, \dots, M_{i-1}, T_1, T_2)$ は将棋プレイヤーが着手を決定する現象の確率分布と考えられる。本稿ではこの現象のモデルを将棋プレイヤーモデルと呼ぶ。



3.3 将棋プレイヤーモデル化の方針

まず、将棋プレイヤーをモデル化の方針を述べる。将棋プレイヤーは全ての手を先読みできるわけではなく、毎回複数の候補から一つの手を選択しなければ

ならない。本稿ではこの選択の理由に注目してモデル化を行う。プレイヤーが着手を選択する際、プレイヤーはその手が持つ性質に注目すると考えられる。例として図3.1の2三にある金を2二に移動する手を挙げる。

この手は“王手を掛ける”と“飛車にあたりを掛ける”という性質を持っている。また、図に表れていない駒の状態によっては他の性質を持っているかもしれない。あるプレイヤーが着手を選択する時、これらの性質のうちどの性質を重要視するかは確率的に決まると考えることにする。また、観測できる着手にはこれら着手の性質情報は含まれていない。よって、本モデルにおいて棋譜は不完全データとなるので、EMアルゴリズムで学習を行う必要がある。

3.4 将棋プレイヤーモデルの概要

まず、本モデルの構築に用いる用語を定義する。

- カテゴリ ある局面について、ある手が持つ性質。プレイヤーが手を選択する意図を表す。
- 概念 プレイヤーがカテゴリを選択する際に用いる情報。

カテゴリの例を以下に示す。

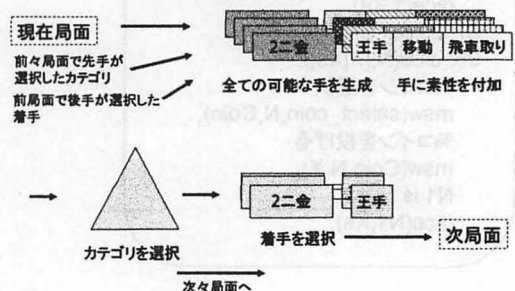
素性名	説明	具体例	説明
capture	駒を取る手	capture(hi)	飛車を取る手
promote	駒を成る手	promote(fu)	歩を取る手
attack	敵駒にあたりを掛ける手	attack(ka)	角にあたりを掛ける手
evade	味方の駒を守る手	evade(ky)	番を守る手
idou	盤上の駒を動かす手	idou	盤上の駒を動かす手
oute	王手を掛ける手	oute	王手を掛ける手

また、概念の例を以下に示す。

名前	概念
root	全ての手
major_decision	詰まず以外の目的の手
ou_anzendo	王の安全に関する手
koma_sontoku	駒の獲得に関する手
charge	攻撃的な手
protect	守備的な手
misc	その他の手

次に、本モデルの概要を図3.3に示す。

図3.3 将棋プレイヤーモデルの概要



本モデルは各局面について以下の情報を入力とする。

1. 現局面の情報
2. 前局面において自分が選択したカテゴリ
3. 対戦相手が直前に選択した着手

本モデルの動作は次のとおりである。

- I. 入力された現局面の情報から選択可能な手の候補を列挙する。
- II. 各手についてその手が持つ全ての性質情報を列挙、付与する。
- III. 現局面におけるカテゴリを選択する。カテゴリの選択についての詳細は後述する。
- IV. 全ての手から選択したカテゴリを持つ手を取り出し、その中から着手の一つを選択する。
- V. 現局面に今選択した着手を反映させ、次局面に移る。

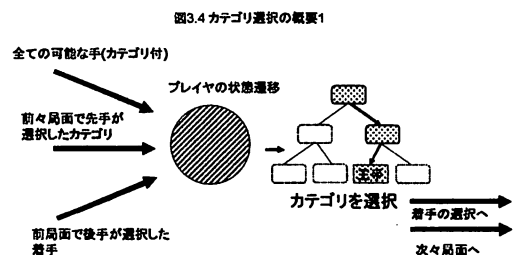
3.5 カテゴリの選択のモデル

図 3.4 がカテゴリ選択の概要である。

3.5.1 プレイヤの状態

まず、プレイヤーの状態について説明する。プレイヤーは次の3つの状態を持つものとする。

1. 攻撃状態
2. 守備状態
3. その他状態



プレイヤーの選択はこれらの状態に依存する。これらの状態はプレイヤーの連続した作戦を表現するために導入する。例として、次の連続した3つの局面を考える。ただし、局面 3.1 と 3.2 は場合分けである。

局面1 先手がある駒にあたりを掛ける。

局面2 後手が駒にあたりを掛ける。

局面3.1 先手が局面 1 であたりを掛けた駒を取る。

局面3.2 先手があたりを掛けられた駒を守る。

敵の駒にあたりを掛けるのは攻撃的な行為であるので、局面 1 後の先手は攻撃状態であるものとする。

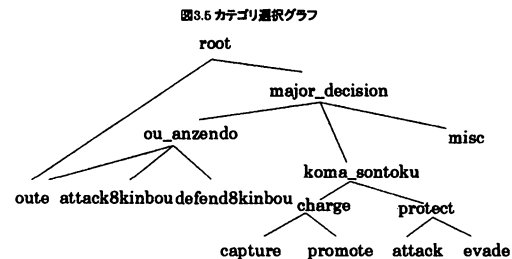
局面 3 で先手は局面 2 における行動に対応するか否かを選択する必要がある。局面 3.1 は先手が後手の行動の影響を受けず、局面 1 における作戦を継続した場合である。この場合、カテゴリ選択時の先手の状態は攻撃状態を維持する。局面 3.2 は後手の行動に対応して、局面 1 における作戦を中断した場合で

ある。この場合、カテゴリ選択時の先手の状態は守備状態になる。

このような現象を表現するため、プレイヤーのカテゴリ選択の前に状態遷移を導入する。状態遷移を行う場合は、相手の着手に対応して自分の作戦を変更することを意味し、状態遷移を行わない場合は、相手の着手を無視して自分の作戦を継続することを意味する。

3.5.2 カテゴリの確率的な選択

カテゴリの選択は図 3.5 のグラフを用いて選択する。このグラフの葉ノードはカテゴリ、その他のノードは概念である。カテゴリの選択は root ノードから開始し、途中のノードで確率的な選択を行いながらグラフを辿ってゆき、辿りついた葉ノードに対応するカテゴリを選択する。



また、この時選択したカテゴリが、攻撃的なカテゴリを選択した場合は攻撃状態、守備的なカテゴリを選択した場合は守備状態、それ以外のカテゴリを選択した場合はその他状態に状態遷移を行い、現局面の作戦を次々局面に伝えることを意味する。

最後に、図 3.6 に将棋プレイヤーモデルの一部を載せておく。

図 3.6 将棋プレイヤープログラムの一部

```

play(Kyokumen, PrevState, EnemyMove, EnemyState, ...)-
%着手を選択する。
think_move(Kyokumen, Move, PrevState, EnemyMove, NewState),
%着手を盤面に反映させる。
update_kyokumen(Kyokumen, Move, NewKyokumen),
%次局面
play(NewKyokumen, EnemyState, Move, NewState, ...).

think_move(Kyokumen, Move, PrevState, EnemyMove, NewState):-
%全ての可能な手を生成する。
generate_all_moves(Kyokumen, AllMoves),
%手に属性を付与する。
analyze_features(AllMoves, FeaturedMoves),
%状態遷移を行う。
change_state1(PrevState, EnemyMove, State),
%カテゴリの選択を行う。
select_category(Kyokumen, State, FeaturedMoves, Category),
%選択したカテゴリに応じて状態遷移を行う。
change_state2(State, Category, NewState)
%着手を選択する。
select_move(Category, FeaturedMoves, Move).

select_category(Kyokumen, NewState, FeaturedMoves, Category):-

%多値スイッチ宣言
values(major_decision(_), [ou_anzen-do, koma_sontoku, misc]).

select_category(major_decision(Kyokumen, FeaturedMoves, State, Category) :-
make_candidate(Kyokumen, FeaturedMoves, Candidate),
%ou_anzen-do, koma_sontoku, miscのいずれかの概念を確率的に選択
msw(major_decision, Candidates, State, Next),
select_category(Next, Kyokumen, FeaturedMoves, State, Category).

```

4 実験

4.1 棋譜データ

本稿では、「将棋倶楽部 24 万局集」[3] の付録 CD-ROM に収録されている棋譜データベースを用いた。各棋譜には棋譜、対局に要した時間、プレイヤー名、対局日時、およびプレイヤーのレーティング情報などが記録されている。今回は、表 4.1 に示した 4 つの棋力それぞれについて、同じ棋力のプレイヤー同士の対局を 100 局ずつ抽出した。

表4.1 棋力とレーティング

棋力	レーティング
アマ4段	2100~2299
アマ初段	1550~1699
アマ5,6級	950~1149
アマ12~14級	250~449

4.2 確率パラメータの学習

上記のデータを訓練データとし、PRISM 処理系の EM 学習を用いて各棋力についての確率パラメータを学習した。また、対数尤度の収束判定閾値 ϵ は 10^{-6} とした。EM 学習における反復回数、収束時の対数尤度、および学習したスイッチ数を表 4.2 に示す。

表4.2 学習結果

棋力	EM反復回数	対数尤度	スイッチ数
アマ4段	420	-3090.725	559
アマ初段	817	-3524.497	561
アマ5,6級	394	-3122.873	553
アマ12~14級	2083	-3199.502	558

4.3 確率パラメータの直接比較

学習されたスイッチの一部を表 4.3、表 4.4、および表 4.5 に示す。

表 4.3 は状態によってプレイヤーの思考が異なっていることを示す例である。このスイッチは、王に関する手、駒のやりとりに関する手、その他の手のいずれを選択するかを決定するものである。この数値は、守備状態では駒のやり取りに関する手を選択する確率が高い等、状態によってプレイヤーの選択が大きく異なることを示している。

表4.3 major_decision(shokuyuu_sente,ou_anzendo,koma_sontoku,misc)

	attack	defence	misc
ou_anzendo	0.431013	0.090220	0.048511
koma_sontoku	0.330667	0.529046	0.197897
misc	0.238320	0.380734	0.753591

表 4.4 と表 4.5 は棋力によって手の選択が異なることを示す例である。表 4.4 は自分が攻撃状態、かつ

相手の直前の手が攻撃的なものであった場合の状態遷移を行うスイッチである。この数値は棋力が高いほど相手の手に応じて作戦を変更する傾向が強いことを示している。

表4.4 change_state(attack,attack)

	4段	初段	5,6級	12~14級
yes	0.622177	0.436761	0.300556	0.220366
no	0.377823	0.563239	0.699444	0.779634

表 4.5 は、攻撃状態の時、王に関する手、駒のやりとりに関する手、その他の手のいずれを選択するかを決定するスイッチである。この数値は棋力が高いほど王に関する手を選択する傾向が強いことを示している。

表4.5

major_decision(shokuyuu_sente,ou_anzendo,koma_sontoku,misc,attack)

	4段	初段	5,6級	12~14級
ou_anzendo	0.483207	0.450347	0.438023	0.431013
koma_sontoku	0.295182	0.280164	0.328279	0.330667
misc	0.221611	0.269489	0.233699	0.238320

4.4 プレイヤーの棋力識別実験

PRISM 処理系には、与えられたデータの生成確率を計算する機能も備わっている。これを用いてプレイヤーの棋力識別実験を行った。

4.4.1 実験の概要

プレイヤー情報の無い棋譜について、この対局を行ったプレイヤーの棋力を識別する。今回は、テストデータの棋譜とプレイヤーの棋力を 5,6 級と 12~14 級に限定して実験を行った。

4.4.2 プレイヤー識別方法

棋譜が既知であるとする。

このときプレイヤーの組の条件付確率

$P(\text{プレイヤー1,プレイヤー2}|\text{棋譜})$ を、5,6 級プレイヤーの組と 12~14 級プレイヤーの組について比較して、値の大きいほうのプレイヤーの組を答える。

$P(\text{プレイヤー1,プレイヤー2}|\text{棋譜})$ を変形すると、

$$P(\text{プレイヤー1,プレイヤー2}|\text{棋譜}) =$$

$$\frac{P(\text{棋譜}|\text{プレイヤー1,プレイヤー2})P(\text{プレイヤー1,プレイヤー2})}{P(\text{棋譜})}$$

となる。

ここで、 $P(\text{棋譜})$ は定数、プレイヤーの組の確率は

一様であると仮定すると、 $P(\text{プレイヤ1,プレイヤ2|棋譜})$ の比較を行うためには $P(\text{棋譜|プレイヤ1,プレイヤ2})$ を比較すればよいと分かる。

4.4.3 実験

節4.1で用意した5,6級同士の対局の棋譜と12~14級同士の対局の棋譜のなかから、パラメータ学習に用いていない棋譜をそれぞれ10局ずつ抽出し、これをテストデータとしてプレイヤ識別実験を行った。また、データスパースネスの問題によって学習されなかったパラメータについては手動で一様分布のパラメータを付与している。

表4.6.5. 6級のテストデータを用いた実験結果

識別の正誤	P(棋譜 12~14級)	P(棋譜 5,6級)
○	1.326014e-146	8.553989e-146
×	3.464655e-151	1.138483e-151
×	3.952928e-159	2.597358e-159
×	5.372780e-156	5.165218e-157
○	2.074154e-143	3.357339e-142
	1.055242e-168	0.000000e+00
○	2.265991e-123	3.688488e-123
×	3.358237e-160	2.768142e-161
×	3.320475e-158	4.890608e-159
×	1.464465e-138	1.029547e-140

表4.7 12~14級のテストデータを用いた実験結果

識別の正誤	P(棋譜 12~14級)	P(棋譜 5,6級)
○	4.595618e-122	9.428926e-124
○	5.793266e-128	2.393013e-131
×	3.824662e-156	1.505177e-152
○	6.872506e-144	1.584584e-144
○	3.527628e-124	6.397613e-125
×	9.713136e-167	2.117463e-166
○	2.423126e-135	4.627068e-138
○	7.385202e-143	2.315896e-146
○	1.701219e-135	2.045888e-136
○	1.665459e-168	5.829593e-170

表4.6は5,6級同士の対局の棋譜をテストデータとして用いた識別実験の結果、表4.7は12~14級同士の対局の棋譜をテストデータとして用いた識別実験の結果である。成績は、5,6級同士の場合は10問中3問正解、12~14級同士の場合は10問中8問正解であり、合計すると20問中11問正解と正解率は5割を超えた。

4.5 考察

確率パラメータ学習実験については、データスパースネスは生じているものの、状態によって値が異なるパラメータや棋力と対応して値が異なるパラメータを学習することができた。特に、`change_state`

のパラメータは上級者ほど相手の手に応じて柔軟な手を選択することを示しており、妥当な値が得られたと言える。

しかし、棋力識別実験では、正解率は5割を超えたもののテストデータの棋譜にかかわらず全体的に12~14級プレイヤの方が確率が大きくなる傾向がみられており、有効な結果が得られたとは言えない。このような結果になった原因はデータスパースネスであると考えられる。本実験においては5,6級プレイヤについては26個、12~14級プレイヤについては23個のパラメータを手動で付与しており、データスパースネスが深刻であると分かる。

5. まとめと今後の課題

5.1. まとめ

本稿では、PRISMを用いて将棋プレイヤモデルの構築を行った。また、公開されている棋譜データを用いて各棋力についてのパラメータ学習実験を行い、得られたパラメータを用いて与えられた棋譜のプレイヤ識別実験を行った。パラメータ学習実験ではプレイヤの状態や棋力に応じて値が異なるパラメータを学習することができた。しかし、プレイヤ識別実験ではデータスパースネスの問題により、有効な結果を得ることができなかった。

5.2. 今後の課題

まず、確率パラメータ学習実験、棋力識別実験の双方においてデータスパースネスが問題となった。この問題を解決するために、今後大規模な訓練データを用いた実験を行ってゆく。

次に、モデルそのものの改良が挙げられる。現状では、本モデルは現実的な棋譜を出力することができない。そこで、本モデルと従来の将棋プログラムの手法とを組み合わせることで現実的な棋譜を出力するモデルの構築する必要がある。この改良を行うことにより、不完全情報ゲームの相手モデル等への応用が期待される。

6. 参考文献

- [1] 久米 宏：将棋倶楽部24万局集，ナイタイ出版(近代将棋) (2001).
- [2] Sato, T and Kameya, Y：PRISM: a language for symbolic-statistical modeling, Proc. of IJCAI'97, pp.1330-1335 (1997).
- [3] 磯部 正幸, 佐藤 泰介：記号的統計モデリング言語PRISMを用いた将棋プレイヤのモデル化 (2002).
- [4] 松原 仁：将棋とコンピュータ，共立出版 (1994).
- [5] 松原 仁 編著：コンピュータ将棋の進歩，共立出版 (1996).

- [6] 松原 仁 編著：コンピュータ将棋の進歩 2, 共立出版 (1998).
- [7] 松原 仁 編著：コンピュータ将棋の進歩 3, 共立出版 (2000).
- [8] 松原 仁 編著：アマ4段を超えるーコンピュータ将棋の進歩 4, 共立出版 (2003).