Vol.2011-GI-26 No.2
2011/7/1

帰納論理プログラミングを用いた棋譜からのルール抽出

力規晃[†]　越村三幸[††]　藤田博[††]　長谷川隆三[††]

これまでコンピュータ将棋を強くする研究は多数行われてきたが，将棋の熟達度に関して棋譜を分析した研究はあまり多くはない．本研究では将棋の熟達度に応じた指し手や駒の利きに関するルールを，帰納論理プログラミングを用いて抽出する．

Rule Extraction from Shogi Game Records Using Inductive Logic Programming

Noriaki Chikara[†] Miyuki Koshimura[††] Hiroshi Fujita[††] and Ryuzo Hasegawa[††]

There are a few studies on analyzing Shogi (Japanese chess) game records focusing on mastery level, while there are a lot of studies for strengthening computer programs playing Shogi. This paper tries to extract rules from Shogi game records using inductive logic programming. We expect that these rules represent relations between moves and effects of Shogi player's and characterize his/her skill.

1. はじめに

コンピュータ将棋の強さは日進月歩で，コンピュータ将棋を強くするために多くの研究や試みが行われてきた．しかし，認知科学的な研究はあまり多くはない．これまで，伊藤らは次の一手に関して発話とアイカメラにより将棋の熟達化と思考過程との関係を明らかにする試みを行っている[1]．この試みの発話内容には指し手に関することだけでなく，周辺の駒の利きを意識した発言もいくつか見られる．

そこで本研究では将棋の熟達の度合いに応じた棋士の指し手と駒の利きの関係を明確にすることを目的として，棋譜データをレーティングに基づいて棋士の棋力で分類し，そして，帰納論理プログラミングを用いて，棋力ごとのルール抽出を試みる．

我々は棋力ごとの指し手と利きの規則性を求めるために帰納論理プログラミングを用いる．これまで詰将棋に関しては帰納論理プログラミングを利用しルール抽出を試みる研究[2]は行われているが，将棋の実際の指し手については帰納論理プログラミングを利用した例はない．帰納論理プログラミングを用いることにより棋譜の規則性を述語論理で記述したルールで得られる．したがって，述語論理の記述性の高さにより，将棋の熟達に関する複雑な規則を得ることが期待できる．

また，得られたルールは棋士の棋力の自動判定等への応用も考えられる．

2. 棋譜データ

本研究では，将棋倶楽部 24[3]の棋譜データベースを利用する．棋譜データを棋士のレーティングごとに，ルール抽出に用いる．

2.1 将棋倶楽部 24

将棋倶楽部 24[3]とは会員数 24 万人(2010 年 5 月現在)の世界最大のインターネット将棋サイトである．

2.2 レーティング

将棋倶楽部 24 では，棋士の棋力の客観的指標としてレーティングを用いている．ユーザーは登録時に自己申告でレーティングを定め，その後は対局で勝利すれば，レーティング値は上がり，負ければ下がる．対局を繰り返すことでユーザーの実力を適切に表すレーティング値になる．

本研究では，表 1 のようにレーティングによって棋譜を初級，中級，上級，有段に

[†] 徳山工業高等専門学校
Tokuyama College of Technology
[††] 九州大学
Kyushu University

分類し，それぞれのルール抽出に用いる．

表 1　レーティングの分類

レーティング	分類
0～549	初級
550～1049	中級
1050～1549	上級
1550～	有段

3. 帰納論理プログラミング

本研究では，ルール抽出のために帰納論理プログラミングを用いる．

3.1 帰納論理プログラミング

帰納論理プログラミング(Inductive Logic Programming)[4]とは一階述語論理に基づいた機械学習の手法である．ILP の一般的な枠組みは，正例と負例，背景知識が節集合で与えられ，

$$\begin{cases} B \not\models E^+ \\ B \cup E^- \not\models \square \end{cases} \tag{1}$$

であるとき，

$$\begin{cases} H \cup B \models E^+ \\ H \cup B \cup E^- \not\models \square \end{cases} \tag{2}$$

を満たす仮説Hを見つけることである．本研究では ILP システムとして Prolog 上で動作する Aleph[5]を用いる．仮説 H は探索により求める．

3.2 帰納論理プログラミングで用いる知識表現

帰納論理プログラミングにおいて，知識は述語論理の形式で表現する．棋譜データから表 2 の述語表現を用いて，棋譜から得られた指した手とその駒，その手の前後関係，駒の利きの関係とその利きの関係にある駒を表現して，背景知識として与える．

ここで，変数 Piece1,Piece2 はある棋譜のある手番での局面のある座標の駒を特定する．

表 2　述語による知識表現

述語による知識表現	意味
mov(Piece1,Promote,Piece2,Tesu)	駒 Piece1 の指し手について，Promote が true の時は成り，false のときは成らない．また，駒 Piece2 を取る手である．Piece2 が null のときは駒を取らない．そして Tesu 番目の手である．
effect(Piece1,Piece2)	駒 Piece1 は駒 Piece2 に利いている
next(Piece1,Piece2)	駒 Piece1 の指手の次の指手の駒は Piece2 である
piece(Piece1,Koma,Suji,Dan)	駒 Piece1 の駒種は Koma であり，Suji 筋 Dan 段にある

また，駒種は表 3 のようなアトムで表現する．アルファベット小文字だけのローマ字で記述された駒種が自分の駒を表し，自駒の名前の最後に‘_’が付いたものは敵駒を表す．

表 3　駒種の知識表現

自駒/敵駒	駒種	知識表現(アトム)
自駒	歩	fu
	香車	kyousha
	桂馬	keiuma
	銀	gin
	金	kin
	角	kaku
	飛車	hisha
	王	ou
	と	tokin
	成香	narikyou
	成桂	narikei
	成銀	narigin
	竜馬	ryuuma
	竜王	ryuou
敵駒	歩	fu_
	香車	kyousha_

	桂馬	keiuma_
	銀	gin_
	金	kin_
	角	kaku_
	飛車	hisha_
	王	ou_
	と	Tokin
	成香	narikyou_
	成桂	narikei_
	成銀	narigin_
	竜馬	ryuuma_
	竜王	ryuou_

3.3 正例と負例

本研究では，正例は表 1 のようにレーティングによって特定のカテゴリに分類された手とし，負例にはそれ以外のカテゴリの手を用いる．

3.4 追加の背景知識

駒種について，自駒と敵駒や大駒と小駒を扱うため図 1 のようなルールを背景知識に追加している．

例えば，次のルール

piece(K,kogoma,S,D):-piece(K,fu,S,D).

は駒 K の駒種が fu（歩）ならば，駒 K の駒種が kogoma（小駒）であることを意味している．このようなルールにより，fu（歩）, kyousha（香車）, keiuma（桂馬）, gin（銀）, kin（金）, tokin（と金）, narikyou（成香）, narikei（成桂）, narigin（成銀）は kogoma（小駒）として扱い，kaku（角）,hisha（飛車）,ou（王）,ryuuma（竜馬）,ryuou（竜王）は oogoma（大駒）として扱う．敵駒も駒種の表現の最後に‘_’を付けて区別しているが，同様のルールが適用される．

また，kogoma（小駒）, oogoma（大駒）は jigoma（自駒）として扱い，kogoma_（敵の小駒），oogoma_（敵の大駒）は tekigoma_（敵駒）として扱うルールも付加されている．

```
piece(K,kogoma,S,D):-piece(K,fu,S,D).
piece(K,kogoma,S,D):-piece(K,kyousha,S,D).
piece(K,kogoma,S,D):-piece(K,keiuma,S,D).
piece(K,kogoma,S,D):-piece(K,gin,S,D).
piece(K,kogoma,S,D):-piece(K,kin,S,D).
piece(K,kogoma,S,D):-piece(K,tokin,S,D).
piece(K,kogoma,S,D):-piece(K,narikyou,S,D).
piece(K,kogoma,S,D):-piece(K,narikei,S,D).
piece(K,kogoma,S,D):-piece(K,narigin,S,D).

piece(K,oogoma,S,D):-piece(K,kaku,S,D).
piece(K,oogoma,S,D):-piece(K,hisha,S,D).
piece(K,oogoma,S,D):-piece(K,ou,S,D).
piece(K,oogoma,S,D):-piece(K,ryuuma,S,D).
piece(K,oogoma,S,D):-piece(K,ryuou,S,D).

piece(K,jigoma,S,D):-piece(K,kogoma,S,D).
piece(K,jigoma,S,D):-piece(K,oogoma,S,D).

piece(K,kogoma_,S,D):-piece(K,fu_,S,D).
piece(K,kogoma_,S,D):-piece(K,kyousha_,S,D).
piece(K,kogoma_,S,D):-piece(K,keiuma_,S,D).
piece(K,kogoma_,S,D):-piece(K,gin_,S,D).
piece(K,kogoma_,S,D):-piece(K,kin_,S,D).
piece(K,kogoma_,S,D):-piece(K,tokin_,S,D).
piece(K,kogoma_,S,D):-piece(K,narikyou_,S,D).
piece(K,kogoma_,S,D):-piece(K,narikei_,S,D).
piece(K,kogoma_,S,D):-piece(K,narigin_,S,D).

piece(K,oogoma_,S,D):-piece(K,kaku_,S,D).
piece(K,oogoma_,S,D):-piece(K,hisha_,S,D).
piece(K,oogoma_,S,D):-piece(K,ou_,S,D).
piece(K,oogoma_,S,D):-piece(K,ryuuma_,S,D).
piece(K,oogoma_,S,D):-piece(K,ryuou_,S,D).

piece(K,tekigoma_,S,D):-piece(K,kogoma_,S,D).
piece(K,tekigoma_,S,D):-piece(K,oogoma_,S,D).
```

図 1　駒種に関する追加のルール

また，盤面を図 2 のように 9 つの領域に分割して扱うため，図 3 のルールを追加する．これらによって，1〜3 の筋は migi（右），4〜6 の筋は naka（中），7〜9 は hidari（左）として扱い，一〜三の段は joudan（上段）,四〜六段は chudan（中段）,七〜九段は（下段）として扱う．但し，後手側の手番の場合は座標を反転させて，先手側に合わせて扱う．

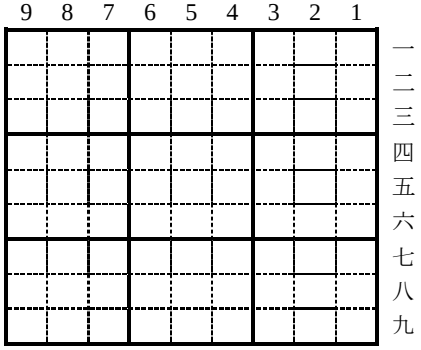

図 2　盤面の領域

```
piece(P,K,migi,D):-piece(P,K,1,D).
piece(P,K,migi,D):-piece(P,K,2,D).
piece(P,K,migi,D):-piece(P,K,3,D).
piece(P,K,naka,D):-piece(P,K,4,D).
piece(P,K,naka,D):-piece(P,K,5,D).
piece(P,K,naka,D):-piece(P,K,6,D).
piece(P,K,hidari,D):-piece(P,K,7,D).
piece(P,K,hidari,D):-piece(P,K,8,D).
piece(P,K,hidari,D):-piece(P,K,9,D).

piece(P,K,S,joudan):-piece(P,K,S,1).
piece(P,K,S,joudan):-piece(P,K,S,2).
piece(P,K,S,joudan):-piece(P,K,S,3).
piece(P,K,S,chudan):-piece(P,K,S,4).
piece(P,K,S,chudan):-piece(P,K,S,5).
piece(P,K,S,chudan):-piece(P,K,S,6).
piece(P,K,S,gedan):-piece(P,K,S,7).
piece(P,K,S,gedan):-piece(P,K,S,8).
piece(P,K,S,gedan):-piece(P,K,S,9).
```

図 3　盤面領域を扱うための追加ルール

4. 実験

本研究では Prolog システムは YAP[6] 6.2.0 を用い，その上で ILP システム Aleph[5] Version5 を実行した．また，棋譜データは将棋倶楽部 24 万局集[7]のデータを用いる．表 1 のレーティングの分類と指した手の駒種ごとにルール抽出の実験を行った．
実験用 PC のスペックは表 4 に示す．

表 4　実験用 PC

CPU	Intel Core i7 2.7GHz (コア数 2)
メモリ	8GB
OS	Windows7 64bit

5. おわりに

帰納論理プログラミングを用いて，将棋の棋譜から棋力に応じた指し手と利き駒の規則性を示したルール抽出の実験を行った．今後，今回抽出したルールが棋士の思考とどのような関係にあるかを検討していくべきだと考えられる．

謝辞　本研究は科研費(21300054)の助成を受けたものである．

参考文献

1)　伊藤毅志，松原仁，グリンベルゲン ライエル: 将棋の認知科学的研究(2)，次の一手実験からの考察，情報処理学会論文誌，Vol.45, No.5, pp.1481-1492 (2004)
2)　Nakano. T., Inuzuka N. Seki H. and Itoh H.: Inducing Shogi Heuristics Using Inductive Logic Programming, Proceedings of the 8th International Workshop on Inductive Logic Programming(ILP '98), LNAI Vol.1446, Springer, pp.155-164 (1998)
3)　将棋倶楽部 24, http://www.shogidojo.com/
4)　古川康一，尾崎知伸，植野研: 帰納論理プログラミング，共立出版 (2001)

5) Srinivasan A.: The Aleph Manual,
http://www.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/ (1999)
6) YAP Prolog, http://www.dcc.fc.up.pt/~vsc/Yap/
7) 久米宏: 将棋倶楽部 24 万局集, ナイタイ出版 (2002).