

4 F-2 帰納学習を使った詰め将棋戦略の獲得

駒沢 寿夫 北村 太路 犬塚 信博 山田 雅之 加藤 昇平 世木 博久 伊藤 英則
名古屋工業大学

1 はじめに

我々はこれまでに詰め将棋を効率良く解く方法を開発してきた [1][2]. [2] では共謀数を利用した並列探索法を提案し, 高い解答能力を実現している. しかし, こうした解法は人間の指し手と大きく異なり, より人に近い思考方法の導入が今後の検討課題となっている. 機械学習の利用もその一つである. そこで本研究では, 手始めとして人がゲームに関して持ついくつかの概念を機械学習の手法により獲得することを試みる.

最近, 帰納論理プログラミング (ILP: Inductive Logic Programming) が機械学習の新たな方法として注目されている. そこで, これを将棋の問題に適用する. これまでもチェスへの ILP の適用が試みられている [3] が, 小規模な終盤への適用であっても必ずしも良い結果は得られていない. 本研究では, 人が将棋に関する概念を獲得するのと同じように, 順に概念を帰納し, この方法の可能性を探る.

2 帰納論理プログラミング (ILP)

ILP は, 与えられた事例を説明する論理プログラムを帰納的に求めることを目的とする. 目標とする述語の正しい事例の集合 (正事例), 誤った事例の集合 (負事例), そして背景知識としていくつかの述語に関するプログラムを与えることにより目標述語の論理プログラムを導出する. 本研究では ILP の手法の一つトップダウンアルゴリズム (図 1) に基づく ILP システム FOIL-I を用いた.

FOIL-I [4] は, Quinlan により提案され広く用いられる FOIL [3] を小数事例のみからの帰納に関して効率良く動作するよう改良したものである. 詰め将棋のような大規模な背景知識を含む問題を扱うため, FOIL-I を以下の点で更に見直す.

1. 広範囲の値を取り得る変数の効率的処理
2. 出力モードの変数に関する制限の緩和
3. 否定リテラルの扱い
4. 大規模背景知識の管理

Acquiring Shogi Knowledge Using Inductive Logic Programming
Hisao Komazawa, Taiji Kitamura, Nobuhiro Inuzuka, Masashi Yamada, Shohei Kato, Hirohisa Seki and Hidenori Itoh.
Nagoya Institute of Technology.
Gokiso-cho, Showa-ku, Nagoya 466, Japan

```

1 theory := 空プログラム
2 remaining := 目標とする述語の正事例
3 While remaining が空でない
4   clause := "R(A,B,...) :-"
5   While clause が R に属さない事例を導出する
6     適当なリテラル L を見つける
7     L を clause の body に加える
8   clause に導出された事例を remaining から除く
9   clause を theory に加える

```

図 1: トップダウン ILP アルゴリズム

3 詰め将棋を学習するための ILP

上述の 4 点について FOIL-I を次のように改良した. 外延的評価から内包的評価へ

FOIL-I ではリテラルが満たす値の組を全て予めリストとして蓄え, 評価時に事例とそのリストとのマッチングを行う. これは, FOIL が関係を外延的に保持し処理することを継承しているためである. しかし広範囲の値を取り得る変数を含む場合ではリストが長大となり効率的処理が行なえない. そこで, そのつど評価するリテラルを展開する内包的評価法に変更した.

出力モードの変数の処理

FOIL-I では, 目標とする述語の変数は全て入力モードとみなしこれらの変数を背景知識の出力モードの変数として用いることを禁じていた. そこでこの制約を緩和し目標とする述語の変数が出力モードならば, その変数を出力するリテラルも学習候補となるようにした. また, 出力モードの述語をより効率良く, 素早く学習するよう, 目標とする述語の出力モードの変数を出力してるリテラルの数に応じて優先的に処理した.

否定リテラルの評価

一般に否定リテラルは肯定リテラルに比べて多くの事例が満たされるため良い評価値を得る. このため否定リテラルに関しては不必要な学習を繰り返すことが多い. そこで, 背景知識の評価時に (重み)* (否定リテラルの数) の重みをつけるようにして否定リテラルの学習の展開をおさえるようにした.

背景知識のライブラリ化

FOIL-I を初めとする多くの ILP システムではこれまで小規模の背景知識のみを想定していたためユーザーにとって知識管理の便利な方法が与えられていなかっ

述語	意味
checkmated(盤)	盤が詰んでいる
checkmate(盤, 持ち駒, 駒1, 駒2)	盤上の駒1を駒2に打つことで詰む
king_threatened(盤)	盤上で玉への効きがある
king_evadable(盤)	盤上で玉が逃げられる
gote_defend(盤)	盤上で後手が王手を防げる
possible_aikoma(盤)	盤上で合い駒が打てる
movable(駒1, 駒2, 盤)	盤上で駒1が駒2に動ける
get_king(盤, 駒)	盤上の駒(玉)を得る
sente(駒)	先手の駒である
gote(駒)	後手の駒である
make_aikoma(座標, 駒)	座標の駒(合い駒)作る
play(盤1, 盤2, 駒1, 駒2)	盤1上の駒1を駒2に打ち, 盤2を得る
change_positon(駒1, 座標, 駒2)	駒1の座標を変えて駒2を得る
tobi_koma(駒)	駒は飛び駒である

表1: 学習対象の述語とその説明

た。そこで、背景知識の定義をライブラリとして管理し、システムには背景知識の名前、モード、引数の数のみを与えるようにすることで大規模な背景知識をより便利に用いられるようにした。

4 将棋知識の帰納実験と評価

表1に学習対象とした述語を示す。述語はグループ1,2,3そして基本述語に分けられる。グループ1は2以下、グループ2は3以下、そしてグループ3は基本述語を使って定義することができる(図2)。実験では、まずグループ3の各述語を基本述語と例から帰納することを試み、その結果を使ってグループ2を更にグループ1を帰納させた。その結果を表2に示す。事例は、正/負事例を各5つずつ与えた。

グループ毎の順次的帰納により14個中12個の述語が正しく帰納されることが確認できる。

学習が失敗したものについては、メモリオーバーによるものと(play)、学習結果の述語の定義が正事例を満たさないもの(gote_defend)がある。

正しく帰納された述語の内4つはリテラル数が人の場合よりも減っている。これは、不要なものが導かれず効率が向上した場合(movable)と、効率のためのリテラルを落してしまい、逆に効率が下がった場合(checkmate, 他)があった。

5 おわりに

表2からも分かるように実用的な時間で、それ程多くない事例数で自動的に学習させることができた。プログラムの効率に関わるリテラルを正しく帰納するための方法は今後の課題である。

述語名	学習時間(秒)	リテラル数	
		人	FOIL-I
checkmated	24	4	4
checkmate	10	5	4
king_threatened	6	6	5
king_evadable	9	4	4
gote_defend	(204)	13,15	(10)
possible_aikoma	125	12	9
movable	2	3,3	2,3
get_king	1	3	3
sente	1	2	2
gote	1	2	2
make_aikoma	1	4	4
play	-	9,4	-
change_position	1	4	4
tobi_koma	1	2,2,2	2,2,2

リテラル数が複数あるものは複数の節からなる述語である
-は学習失敗、()は帰納されたが不適切な場合

表2: 詰め将棋の述語の学習結果

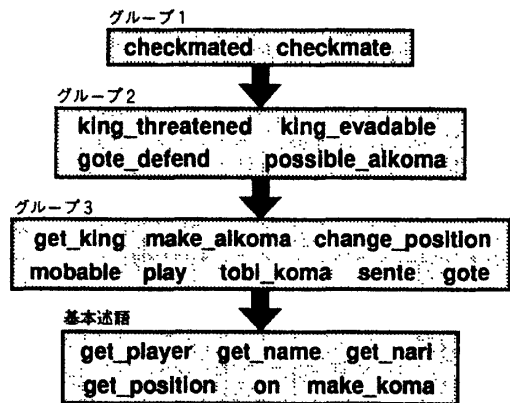


図2: 述語の階層

参考文献

- [1] 笠田, 山田, 松波, 世木, 伊藤: 詰将棋におけるゲーム木の並列探索とその評価, 情報処理学会論文誌, Vol. 36, No. 11, pp. 2531-2539, 1995
- [2] T.Kitamura, S.Kato, M.Yamada, H.Seki, H.Itoh: "A Parallel Search Method Based on Conspiracy Number and its Application to a Tsume-Shogi Program", Proc.6th Parallel Computing Workshop, 1996
- [3] J.R.Quinlan, R.M.Cameron-Jones "Induction of Logic Programs:FOIL and Related Systems", New Generation Computing, 13, pp. 287-312, 1995
- [4] N.Inuzuka, M.Kamo, N.Ishii, H.Seki and H.Itoh: "Top-down Induction of Logic Programs from Incomplete Samples" Proc. 6th Int'l Inductive Logic Programming Workshop, pp. 119-136, 1996.