

3D-6

帰納推論のゲームの学習への応用

毛利 隆夫 田中英彦

{mohri,tanaka}@mtl.t.u-tokyo.ac.jp

東京大学 工学部

1 はじめに

ゲームをコンピュータ上で実行することはAIの分野では古くから研究されてきた分野であり、現在も活発な研究が行なわれている[Abr89]。また一方で事例ベース推論と呼ばれるような、過去の推論結果を蓄えておいて、新しい問題に遭遇した時にそれらを利用して、推論効率を向上させようという研究が最近盛んになってきているが、そこでは事例の格納方法や特徴抽出法などが問題となっている[小林91]。

本研究では探索時の枝刈りのために対戦中の探索結果を事例を利用することを考え、その手法として帰納推論を用いることについて考察する。その際今までに提案された帰納推論手法の問題点を指摘し、要請される機能拡張について考察を加える。

2 ゲームのモデリング

本研究では二人対戦型ゲームの一つである、五目並べを例にとって考察を進めていくことにする(簡単化のため盤面の大きさを 7×7 (本来は 15×15)とし、三々、四々、長連などの禁じ手は考えない(図1))。ゲームプログラムは基本的には α - β 法を用いた探索主導型のプログラムを考えるが、より深いノードを読む前に、過去の事例を用いて有望な差し手のみを選択する機能を持たせることにする。

事例は、

事例 = 〈 盤面, 最良手 〉

の形で蓄えるものとする。通常事例を格納する際には盤面そのままの形ではなく、

事例 = 〈 盤面から抽出された特徴, 最良手 〉

とするところであり、この特徴の決め方が事例ベース推論での選択点の一つであった。帰納推論を用いる場合には特徴に相当するような述語をあらかじめ与えておく必要があるため、特徴の選択枝が固定化される点を解決することはできていない。しかし、よりプリミティブな述語を与えて置くことによって特徴を問題に即したように構成できる余地は残されている。今回は盤面から特徴を抽出し最良手を導き出す過程を帰納推論部に任せることにする。

3 帰納推論の手法

帰納推論とは、与えられた正負の事例から、それらを説明するような一般的な記述を得るような推論操作のことである。

⁰Application of Inductive Reasoning to Game Learning
Takao MOHRI and Hidehiko TANAKA
the University of Tokyo

帰納推論システムには、述語の真である例(quicksortの例では qsort([3,1,2], [1,2,3]), ...) と偽である例(qsort([3,2,1], [1,3,2]), ...) が与えられ、結果を出力するための語彙にあたる述語(partition(...), append(...)) が過不足なく与えられる。システムの解くべき問題は、与えられた正の例を満たし負の例を満たさないような概念を記述する節を導き出すことである。

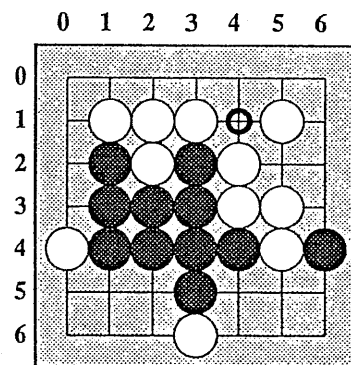
帰納推論の分野では、最近効果的なヒューリスティクスを用いたり、推論するクラスを制限することによって推論効率を大幅に向上させたシステムがいくつか発表されている。

• Foil

Foil [Qui90] はトップダウンに仮説空間を探索していくプログラムで、情報量(entropy)を計算してノードを選択するヒューリスティクスを用いている。Foilではまず訓練集合(training set)という、正の例/負の例が混ざり合った集合を用意しておき、探索木を一段深く進む際に最も情報量が小さくなるようなノード、つまり訓練集合の正負の例を最も良く弁別するようなノードを選択していく。

• Golem

Golem [MF90] は lgg(least general generalization) [Plo71] を用いてボトムアップに仮説空間を探索していくプログラムである。従来の lgg では出力される節の長さが、正/負の例の数の指数オーダーで増加してしまふ欠点があったが、Golemでは出力される節のクラスに対して制限を加えることによってこの問題を解決している。



次の手番:

黒

最高評価値:

(4, 1)

図1: 五目並べの盤面例

4 帰納推論部に要請される機能

帰納推論部の入力には盤面そのもので、出力は最良手であるとする。事例は時間の余裕がある時点で、通常の探索によって得

られた(盤面, 最良手)の組を用いることにする。例えば, 図1での最良手は探索の結果(4,1)だから, 与える事例は

```
board-candidate([s,s,s,s,s,s,s,
                 s,w,w,w,s,w,s,
                 s,b,w,b,w,s,s,
                 s,b,b,b,w,w,s,
                 w,b,b,b,b,w,b,
                 s,s,s,b,s,s,s,
                 s,s,s,w,s,s,s],[4,1]).
```

(但し s=space, w=white, b=black.)

となる。帰納推論プログラムには, このような事例が数多く蓄えられる。また, プログラムにはあらかじめ

```
five(C,P1,P2,P3,P4,P5). four(C,P1,P2,P3,P4).
three(C,P1,P2,P3). two(C,P1,P2). one(C,P1).
nextto(P1,Dir,P2).
```

(但し $P_i=[X,Y]$, Dir は周辺の8方向, $C=s,w, \text{ or } b$)

といった述語を語彙として与えておく。以上のように設定したゲームの学習という例題は, 今まで Foil や Golem に与えられてきた例題とは幾つかの点で異なっており, そのまま適用してもうまくいかない。それらの異なる点を以下に挙げる。

- 理論名辞 (theoretical term) が含まれている。
与えた述語すべてが使用されるとは限らない。出力に関係のない述語が含まれているわけで, 推論効率が低下するおそれがある。
- LHS(left-hand side) の変数が多い。
Foil では最初に LHS の変数の取り得るすべての場合に關して訓練集合を作成するが, 五目並べの場合には石の置き場所が 15×15 箇所あり, それらが白・黒・空間の3通りの値を取り得るので, LHS 全体の場合の数は $3^{15 \times 15} \times (15 \times 15) \simeq 10^{109}$ となり, とても扱える大きさではなくてなってしまう。
- 事例によって変わる知識 (派生知識) の扱い
たとえば, `point_color(Point,Color)` という, ある場所と色の関係を表す述語を考えてみると, これは事例によって真偽が変化する述語である。つまりある事例では `point([1,1],w)` は真であるが, 他の例では偽であるかもしれない。このような事例が決まらないうちで決定できない知識のことを派生知識と呼ぶことにすると, 派生知識自体は当然プログラムの持つ背景知識として含めることができない。また, LHS と派生知識との関係をあらかじめ列挙しておき, グランドな述語として背景知識に含めておくのは困難である。例えば, `point_color(Board,Point,Color)` という述語を考えて, `point_color([s,w,b,..],[0,0],s),...` といった, 変数のすべての場合に対応するようなグランド項を背景知識に蓄えておけば従来の方法をそのまま利用することは可能であるが, その際にはやはり 10^{109} 個のグランド項が必要になり非現実的である。

5 帰納推論手法の拡張の検討

Foil を拡張して派生知識を扱うようにするのは困難であるが, それは Foil の情報量の計算が原因である。Foil では訓練集合がある節を満たすか満たさないかで分割し, 分割後の正負の集合の要素数から情報量を計算するが, 派生知識を扱うようにすれば事例によって節を満たすか否かが異なってくるため, 情報量

の計算の定義を拡張する必要がある。また, 情報量が計算できたとしても, 幾通りもの訓練集合が前段階においてできており, 次の段階で用いる訓練集合をどう決めていいのかが良く分からない。

一方 Golem を拡張する場合には, Foil の時のように項数が爆発するようなことはないので, 少し簡単になる。Golem の場合仮説を求めるプロセスは, まず個々の事例 e_i に対して背景知識 K から仮説 C_i を下のように導き出し,

$$\begin{aligned} K \wedge C_i &\vdash e_i \\ C_i &\vdash K \rightarrow e_i \\ &\vdash C_i \rightarrow (K \rightarrow e_i) \\ &\vdash C_i \rightarrow (\overline{K} \vee e_i) \end{aligned}$$

仮説 C を $C = \text{lgg}(C_1, C_2, \dots, C_n)$ として求めるものであった。ここで, 派生知識 Kd_i は各事例毎に K と e_i から計算された後は K と同様に扱えて,

$$\begin{aligned} K \wedge Kd_i \wedge C_i &\vdash e_i \\ &\vdash C_i \rightarrow ((K \wedge Kd_i) \rightarrow e_i) \\ &\vdash C_i \rightarrow (\overline{K} \vee \overline{Kd_i} \vee e_i) \end{aligned}$$

のように処理していけばいい。ただ, Kd_i の部分はすべての i において正しいわけではないので, 各仮説 C_i 同士の lgg を取る際に, 任意の2項を組み合わせることができなくなる。この部分はさらに検討する必要があるだろう。

6 おわりに

本研究では事例を用いたゲームの学習の方法として, 帰納推論の手法を用いること検討した。その結果現在知られている帰納推論の枠組を拡張する必要があることが分かった。また, 拡張には Foil のようなトップダウン型の探索プログラムではなく, Golem のようなボトムアップ型の方が適していることも分かった。また今後は拡張した帰納推論の手法を実装し, ゲームの学習に応用していきたい。

謝辞

筆者に Foil, Golem を使用するきっかけを与えて頂き, またいくつかの質問に答えて筆者の理解の手助けをしてくださった東京工業大学志村研究室の沼尾正行, キッスィリクン・ブンサーム両氏に感謝します。

参考文献

- [Abr89] Bruce Abramson. Control strategies for two-player games. *ACM Computing Surveys*, Vol. 21, No. 2, pp. 137-161, June 1989.
- [MF90] Stephen Muggleton and Cao Feng. Efficient induction of logic programs. In *ALT 90*. OHM sha, 1990.
- [Plo71] G.D. Plotkin. *A futher note on inductive generalisation*, volume 6 of *Machine Intelligence*, pp. 101-124. Elsevier North-Holland, New York, 1971.
- [Qui90] J.R. Quinlan. Learning logical definitions from relations. *Machine Learning*, Vol. 5, pp. 239-266, 1990.
- [小林91] 小林重信. 事例ベース推論の研究課題. 情報処理学会研究会報告, Vol. 91, No. 16, pp. 29-38, 1991.