4Q−10　行動履歴からの事例の自動抽出と帰納推論への適用

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

最近帰納推論に関する研究が盛んであるが，多くの場合，正／負の事例は前もって人間が準備し，与えているのが現状である．現実的な問題に帰納推論を応用しようとした際，事例を自動的に抽出し，その正負を何らかの方法で決定する手法が重要になると思われる．

本研究では，簡単な障害物回避ゲームを例題にとり，ゲームでの自分の行動の履歴から，まとまった意味のある部分を取り出す手法，すなわち事例を自動的に抽出する方法について考察する．

また，本研究においては，履歴と事例とルールの間には，興味深い循環的な関係がある．すなわち，履歴から事例を取り出し，その事例からルールを導き，そのルールによって行動することで履歴が生成され，またその履歴から事例が抽出される，といった関係がある．

2　例題：障害部回避ゲーム

本研究では，図1のような障害物回避ゲームを例題とした．このゲームは盤面の右下隅にいるプレーヤーが，左上隅にいるターゲットのところまで，途中の障害物を避けながら，なるべく少ない回数で移動するという，簡単なゲームである．プレーヤーは上下左右の4方向に移動可能で，またいつでも盤面全体に関する情報が得られる(つまり，壁で隠れている場所が見えなくなることはない)．プレーヤーには，自分とターゲットとの距離(障害物の妨害があるので，最少移動回数だとは限らない)が与えられて，これを自分の位置の評価関数として用いる．なお，この問題を解くための背景知識としては，表1のような述語を使用した．

```
greedy_dir_x(+PointFrom,+PointTo,-Dir)
greedy_dir_y(+PointFrom,+PointTo,-Dir)
dir_r(+Dir,-DirRight)
dir_l(+Dir,-DirLeft)
dir_b(+Dir,-DirBack)
point_dir_point(+PointFrom,+Dir,-PointTo)
can_move_point(+Field,+Point)
cannot_move_point(+Field,+Point)
```

表 1: 帰納推論で用いる背景知識

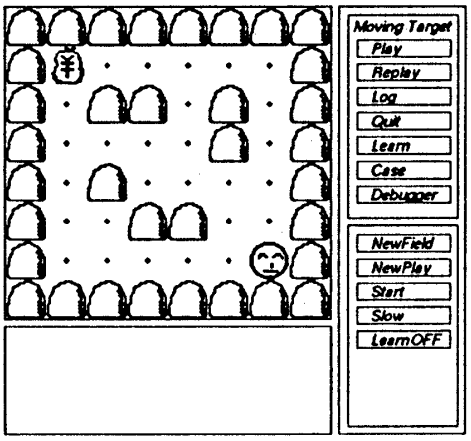

図 1: 障害物回避ゲーム (A)

3　帰納推論

帰納推論部は，基本的にはFOIL [Qui90, Qui91] のアルゴリズムを使用したが，いくつかの改良を行なった．ひとつは，背景知識をground literalで持つ(外延的に定義する)ことは行なわずに，計算式を用いて定義(内包的に定義)しておき，知識が必要になった時点で，それを計算するようにした点である．表1のFieldという変数は，盤面上のどの位置に障害物があるかによって，非常に多くの場合を取り得るために，そのすべての場合に対応した背景知識をground literalの形であらかじめ用意しておくのは，現実的に不可能なのである([毛利91])．

また，lhsの出力となる変数は，rhsのどれかの述語の出力となっていなければならないという制約を課した．これは，生成されるルールによって，lhsの出力となる変数の値を束縛する必要があるからである．

4　行動履歴からの事例の自動抽出

人間は，過去の自分の行動を振り返り，反省して，次の行動を改善するような知見を得ることを日常的に行なっている．将棋などのゲームを行なう場合でも，ゲームが終了した時点で感想戦が行なわれることが，しばしばある．本研究では，ゲームの実行で得られた履歴の中に，以下のような部分があれば，その部分は次の行動を決める際に参考になると考えて，事例として抽出することにした．

1. success
   ルールが出力した方向への移動が成功したとき，その移動を正事例とする．

Case Extraction from History and its Application to Inductive Reasoning
Takao MOHRI and Hidehiko TANAKA
the University of Tokyo

2. failure
   ルールが出力した方向への移動が失敗した場合，その移動を負事例とする．
3. choice point
   choice point(後述) の間の行動列を事例とし，両 point の評価関数の差によって，事例の正負を決める．
4. loop
   同じ場所に戻ってしまうような行動列は，負事例であるとする．

事例は番号の大きいほど優先順位が高いものとし，同じ順位ならば新しい事例の方が優先されるものとした．

4.1 choice point

choice point は，「判断に悩む点」を意味している．ある場所で，いくつかの選択枝のうちどれが最善なのか，わからなかったとする．そこで，選択枝のうちの一つを選んで実行してみると，つぎに悩む時までの行動は一本道になる．したがって，最初に悩んだ時の選択の結果の結果は，少なくとも次に悩む時点，つまり次の choice point まで影響していることになる．そこで，本研究では，choice point 間を事例として，2 点間で評価値が向上しているのならば正事例とし，等しい，もしくは悪化している場合には負事例であるとした．

障害物回避ゲームでは，すべてのルールが次に移動すべき方向を決定できない場合に，止むを得ずランダムな方向に移動する．このようになった場合を choice point であるとした．

5 実験結果

学習の成否は，盤面の障害物の配置によって大きく異なってくる．障害物が密に並んでいて迷路のようになっている場合や，長い壁をなして進路を妨害している場合には，迷路探索アルゴリズムや，大きく迂回することを学習する必要があり，帰納推論を直接的に用いた本研究では，かなり学習が困難であった．

実験は，図 1,2,3の 3 つの盤面において行なった．プレーヤーは，スタートする時点では何もルールを持っていないので，最初はランダムに行動するが，途中で事例が一定個数蓄積できたところで帰納推論を行なってルールを生成し，それ以降はそのルールが出力する方向に移動し，また事例がたまると帰納推論によってルールを作成する．これをゴールにたどり着くまで繰り返すのである．

```
player_dir(Field,Player,Target,Dir) :-
        greedy_dir_y(Player,Target,Dir).
player_dir(Field,Player,Target,Dir) :-
        greedy_dir_x(Player,Target,Dir),
        point_dir_point(Player,Dir,Next),
        can_move_point(Field,Next).
```

表 2: 盤面 A で学習されたルール

図 1のような盤面は簡単に学習できて，表 2 のようなルールが学習された．一方，図 2のような盤面は，幾分学習が難しい．表 2 を見れば分かるように，まず y 座標を合わせた後に，x 座標を合わせるように移動するが，そのままでは途中の障害物に移動を阻止されてしまい，その手前を行ったり来たり振動してしまうのである．振動は 4. の loop によって負の事例として抽出されるので，学習後には振動は回避される．また，図 3は図 2よりもさらに難しく，大きく迂回する必要がある．

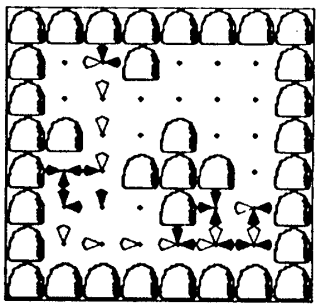

図 2: 浅い凹のある盤面 (B)

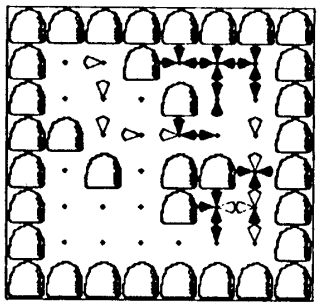

図 3: 深い凹のある盤面 (C)

実験した結果を下表に示す．図 2, 3の中で，白い矢印はその場所からその方向に移動するのが正の事例であることを，黒い矢印は負の事例であることを示している．凹の部分が負事例で満たされているのが分かる．

盤面	A	B	C
帰納推論を行なった回数	2	3	6
帰納推論に要した時間 (sec)	8.7	324.4	442.4
得られた正／負事例数	13/8	12/19	11/24
得られたルールの数	2	6	6

6 おわりに

行動履歴から事例を自動的に抽出し，その事例を用いて帰納推論を行ない，次の行動を決定するルールを得る方法について考察し，いくつかの盤面で実験を行なった．今後の課題としては，より困難な盤面での学習を行なう方法の工夫などが挙げられる．

参考文献

[Qui90] J.R. Quinlan. Learning logical definitions from relations. *Machine Learning*, Vol. 5, pp. 239–266, 1990.

[Qui91] J.R. Quinlan. Determinate literals in inductive logic programming. In *Proc. of 12th International Joint Conference on Artificial Intelligence*, pp. 746–750, 1991.

[毛利 91] 毛利隆夫　田中英彦. 帰納推論のゲームの学習への応用. 第 43 回全国大会講演論文集 (3), pp. 11–12, 1991.