

ILP に基づくトランプのルール抽出

大谷研究室

1372123 望月志穂

1. 研究の背景と目的

トランプを使ったカードゲームは、アプリケーションやインターネット上のコンテンツとして多く存在し、手軽に遊ぶことができる。その際、プレイヤーの人数が足りない場合はコンピュータが対戦相手となる。コンピュータは自分の手番になると、あらかじめプログラミングされた手順に従って次に打てる手を探し、それぞれの手を評価して次の行動を決定する。したがって、コンピュータはある特定のゲームで対戦相手になれても、別のゲームで対戦相手になれるとは限らず、同じゲームでもルールが少し違うと対応できない。遊び相手としてのコンピュータが発展するには、コンピュータが未知のルールを学習し、多くの遊び方に対応することが必要である。本研究の目的は、さまざまなゲームでコンピュータに人の対戦相手をさせることである。本研究ではトランプの大富豪を採り上げ、ゲーム中に行われた手からルールを抽出し、生成されたルールから出せるカードを判断するゲームシステムを構築する。

2. ゲームシステムの概要

ゲームシステムには、ルール抽出モードとゲームモードがある。ルール抽出モードでは、人同士で大富豪をプレイした際のゲーム中に行われた手を入力させ、帰納論理プログラミング(ILP; Inductive Logic Programming) でルールを抽出する。ILP とは背景知識と正例、負例を用いた機械学習手法である。本研究では負例は使用せず、ゲーム中に行われた手を正例、トランプのマークや数字などの情報を背景知識とする。ルール抽出に用いる背景知識と正例は、論理型のプログラミング言語である Prolog で表現する。本研究で用いる知識表現を表 1 に示す。Prolog で記述された背景知識と正例から、ILP に基づいて帰納推論を行う Progol を使用し、ルールを取得する。

ゲームモードでは、人がコンピュータと大富豪をプレイすることができる。コンピュータは自分の番になったら手札から順番にカードの情報を取得し、出せるものがあればカードを出して番を終了する。コンピュータが手札から出せるカードを探索する際は、判別したいカードをルール生成時に入力した正例と合わせて Progol を実行し、生成されるルールの違い、あるいは生成されるルールの有無によって次の手を決定する。

表 1 背景知識と正例に用いる知識表現

3. 評価方法と結果

大富豪では、カードが示す数によって順に強さが決まっており、持っていたカードを一番先にすべて出し終えたプレイヤーが勝ちとなる。基本的なゲームの流れ

述語論理による知識表現	意味
color_is (Mark, Color).	Mark の色は Color である。
strong_than (Number1, Number2).	Number1 は Number2 より強い。
mark_is (Card, Mark).	Card のマークは Mark である。
number_is (Card, Number).	Card の数字は Number である。
take_one (Card1, Card2).	Card1 の上に Card2 を出す。

は以下のとおりである.

- ① カードを全員に配る. 最初にカードを出すプレイヤーを決める.
- ② プレイヤは順番に, 前に出されたカードよりも強いカードがあれば出す. カードを出さずにパスしてもよい.
- ③ 一番強いカードを出したプレイヤーが次の最初のカードを出し, ②へ戻る.

大富豪には“前に出されたカードと同じマークのカードしか出せない”などのルールがあり, 本研究では以下に記した 3 通りの大富豪の遊び方について, 生成されたルールとゲームモードでコンピュータが出した手の正誤からゲームシステムの実用性を評価する. また, カードは 1 枚ずつ出すものとする.

- (I) 前に出たカードよりも数字が大きいカードは出せる.
- (II) 前に出たカードより数字が大きい, かつマークが同じカードは出せる.
- (III) 前に出たカードより数字が大きい, かつマークの色が同じカードは出せる.

1 つの遊び方について人同士で 3 回ずつゲームを行い, ルールを抽出した結果, (I)と(II)では以下のルールが得られ, (III)では 3 回行ったゲームごとに異なるルールが生成された.

- (I) `take_one(A,B):- number_is(A,C), number_is(B,D), strong_than(D,C).`
- (II) `take_one(A,B):- mark_is(A,C), mark_is(B,C), number_is(A,D),number_is(B,E),strong_than(E,D).`
- (III-1) `take_one(A,B):- mark_is(A,C), mark_is(B,C), number_is(A,D),number_is(B,E), strong_than(E,D).`
`take_one(A,B):- mark_is(A,C), mark_is(B,D), number_is(A,E), number_is(B,F), color_is(D,G),`
`color_is(C,G), strong_than(F, E).`
- (III-2) `take_one(A,B):- mark_is(A,C), mark_is(B,C), number_is(A,D),number_is(B,E), strong_than(E,D).`
- (III-3) `take_one(A,B):- mark_is(A,C), mark_is(B,D), number_is(A,E), number_is(B,F), color_is(D,G),`
`color_is(C,G), strong_than(F,E).`

表 2 はルール抽出モードで得られた上記のルールごとにゲームモードで大富豪をプレイした際の, コンピュータが 1 ゲームで行った手番の数である. 表 2 における妥当なパスとは, コンピュータが出せるカードを持っていなかった場合に行ったパスである. また, 不必要なパスとは, 検証した遊び方では出せるカードはあったが, 生成されたルールからコンピュータが出せるカードはないと判断したパスの数である. (III-2)の遊び方については(II)と同じルールが生成されたため, ゲームは行わなかった.

4. 考察

表 2 遊び方ごとの 1 ゲームにおけるコンピュータの行った手

(III)の遊び方について, ゲームによって生成されるルールが違う要因として, Progol では正例の初めから順に推論を開始するため, 人同士で行うゲームの最初の 1 手目に多少依存すること

	ルールに合った手	妥当なパス	不必要なパス	誤った手
(I)	12	13	0	0
(II)	15	10	0	0
(III-1)	14	6	5	0
(III-3)	9	2	14	0

が挙げられる. また, ゲームシステムにおいて, 不必要なパスが多くなったのは, 背景知識の順番によって生成されるルールの並びに違いが出るからである. したがって, 本研究で作成したゲームシステムでは, ゲームのルールや, 正例の書き方によって精度に違いが出るといえる. また, 本研究ではカードを 1 枚ずつ出す遊び方に限って研究をしたが, カードを同時に出す手にも対応することが必要である.