

# チーム keioailab 人狼知能アルゴリズムの概要説明

長谷川 孔明(慶應大)<sup>1</sup>, 堀元 見(慶應大)<sup>2</sup>

## 1. はじめに

チーム「keioailab」では、人狼と狩人の2種類の役職の人狼知能エージェントを作成した（他の役職はデフォルトのエージェントのままである）。エージェントの作成は役職ごとに分担して行い、人狼は堀元が、狩人は長谷川が作成した。

人狼エージェントは、Q学習を用いて各エージェントの「口数」から噛む（襲撃する）相手を判断するアルゴリズムを考案した。

狩人エージェントは、隠れマルコフモデルを用いて「占い師」や「ニセ占い師」といった役職を推測し、護衛先や投票先の判断に用いるアルゴリズムを考案した。

## 2. 人狼の噛み先の指定について

人狼は、噛む相手を「口数」から判断する。

全ての敵エージェントがこれまでに発言した回数を記憶しておき、その回数から、噛む利益が最も大きいと考えられる相手を判断する。

考えられる「噛む利益」の算出のためには、機械学習を利用した。

学習の流れは以下の通りである。

口数が  $n$  の相手を噛む際に見込まれる利益を  $Q(n)$  とする。

- ①  $n = 1 \sim 1000$  として、 $Q(n)$  の値を初期値としてランダムに与えた。
- ② 噛む相手を指定するフェイズになる度に、今のラウンドにおける全てのエージェントの口数を確認し、 $Q(k)$  の値が最大になる口数  $k$  を探す。
- ③ 口数  $k$  のエージェントを噛む。
- ④ ゲーム終了まで①から③を繰り返し、最後に勝敗に応じて  $Q$  の値を更新する。勝利ならば実際に噛んだ相手の全ての  $Q(k)$  を +1 とし、敗北ならば全ての  $Q(k)$  を -1 とする。

これにより、勝敗にひも付けられた「噛む利益」を算出した。なお、本プログラムでは 400 回のゲームから学習させたデータを用いた。

---

<sup>1</sup> komei@ailab.ics.keio.ac.jp

<sup>2</sup> horimoto@ailab.ics.keio.ac.jp

### 3. 狩人の占い師・ニセ占い師の推測

狩人は、護衛先や投票先を決める際に重要となる本物の占い師か人狼が偽ったニセの占い師かを推測するために、隠れマルコフモデルを利用した（図 1 参照）。隠れマルコフモデルは、観測ができない（隠れ）状態を持った確率モデルであり、観測可能な情報から隠れ状態を推測するために用いることができる。

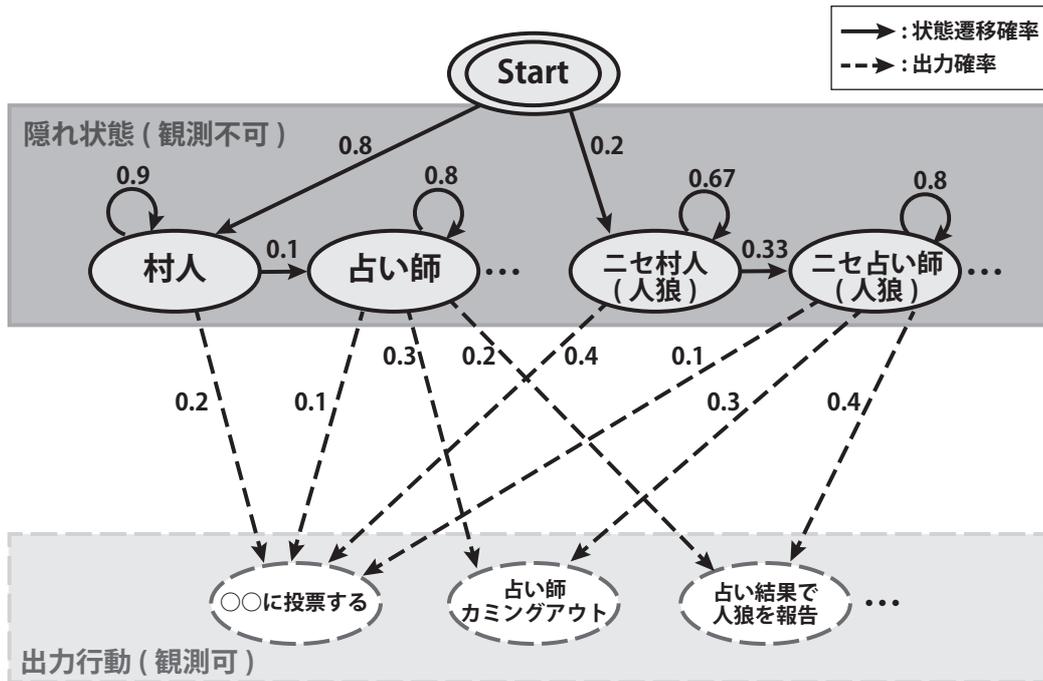


図 1 役職推測のための隠れマルコフモデルの概要

隠れ状態として役職を含んだエージェントの状態（例えば、村人や人狼、人狼がなりすましているニセ占い師など）を用意し、観測可能な出力としては、対戦中の発言や起こった事象（例えば、「〇〇に投票する」「自分が占い師だと名乗り出る」などの発言や、襲撃された、襲撃されなかったなどの事象）を用意した。

隠れマルコフモデルでは、隠れ状態が移り変わる確率（状態遷移確率）や、ある状態にてどの行動を出力しやすいかという確率（出力確率）をモデルパラメータとして設定する必要がある。パラメータは、確率を計算できる項目（15人中3人が人狼の役職に割り振られるなど）についてはその確率を利用し、それ以外の部分については設計者の人狼ゲームに対する直感（ニセ占い師は本物の占い師よりも「占いの結果、〇〇は人狼である」と報告しやすいなど）で決定した。

対戦時には、発言や事象が観測される度に、前向きアルゴリズムを用いて各エージェントがどの隠れ状態でありそうかの確率を更新した。その確率を元に、占い師の確率が高いエ

エージェントを護衛したり，人狼の確率が高いエージェントに投票したりといった行動を決定した。

今回の実装では状態遷移確率などのパラメータの多くを直感で決定しているが，実際の対戦データに基づいて統計的に出した確率を利用するといった，根拠のあるパラメータ設定に改善できると考えられる。また，隠れマルコフモデルの設計時に用意する隠れ状態の種類や，観測できる出力の種類によって推測の精度や推測できる状態の粒度なども大きく変わってくる。