

JuN1Ro アルゴリズム概要

チーム : JuN1Ro

- 実行手順

Player クラス名 : JuN1roPlayer

- コンセプト

GAT2016 ミニ大会優勝のチーム cash さんの考え方をベースに、陣営推測テーブルというものを作成し、利用しています。また、人狼・狂人・占い師について複数のストラテジーを用意し、それぞれに異なる更新規則を持つ陣営推測テーブルを実装しました。つまり、上記役職については異なるアルゴリズム+異なる陣営推測テーブルを持つ、複数のエージェントが存在します。

この複数のエージェントに実際のゲームで同時に行動させ、これらのエージェントの多数決で JuN1ro の行動を決めています。さらに各エージェントの各メソッドは重みを持っており、これらの重みに基づいて重み付きの多数決で行動が決まります。

本来重みの学習も可能で、学習予定でしたが、100 試合中での学習は困難であったため、実際に使用したエージェントは初期値（ランダム）のままです。

いわゆるアンサンブル学習によるエージェントです。

- 基本戦略について

- 村人陣営

陣営推測テーブルに沿って最も人狼の可能性が高いプレイヤーに投票します。

役職持ちの能力行使対象も基本的に陣営推測テーブルを用いて選択しています。

基本的に投票時や発言時に他者に便乗しません。自らの推測に基づいた行動をとります。

- 人狼陣営

陣営推測テーブルに沿って最も人狼の可能性が高いプレイヤーに投票します。ただし、人狼陣営の人狼の可能性は予め 0.5 倍しておきます。このため、場合によっては味方に投票するなどの行動ができます。

アルゴリズムによっては投票時や発言時に他者に便乗します。

- 各種アルゴリズム

- 陣営推測テーブル

1. 概要

各エージェントの陣営を推測するための基礎アルゴリズムについて説明します。コンセプトに記載の通り、GAT2016 ミニ大会優勝チーム cash さんのアルゴリズムを参考にさせていただきました。

各エージェントはゲーム開始時に陣営推測テーブルと倍率テーブルという 2 つのテーブルを

作成します。各テーブルのサイズは、エージェントがそれぞれ、人狼、狂人、村人陣営のいずれかであるパターンの全組合せ (5460 通り) と等しくなっていて、初期値は 1.0 となっています。倍率テーブルは陣営推測テーブルの値を変更するためのテーブルであり、エージェントは倍率テーブルを複数保持しています。

2. 倍率テーブルの更新

エージェントの各アクションの最初に、特定の条件（エージェントが新たに CO する、エージェントが占いや霊能結果を言う、投票結果が出る等）を満たした場合、その時の状況や満たした条件ごとに設定した値（この値を更新倍率と呼ぶことにします）に従って、倍率テーブルの値を更新します。更新される倍率テーブルはその更新タイミングによって異なります。

3. 陣営推測テーブルの更新

倍率テーブルが更新された直後に、その倍率テーブルを用いて陣営推測テーブルを更新します。陣営推測テーブルの値とそれに対応した倍率テーブルの値の乗算した結果を新たな陣営推測テーブルの値とします。更新後、陣営推測テーブルは最大値が 1 となるように規格化し、倍率テーブルは陣営推測テーブルの更新に用いたものは初期化します。

4. 陣営の推測 あるエージェント

A の人狼らしさは (A が人狼である場合の陣営推測テーブルの値の和) / (陣営推測テーブルの値の総和) で求められます。狂人らしさ、村人陣営らしさもそれぞれ同じ方法で求められます。陣営推測テーブルは各エージェントが人狼、狂人、村人陣営のいずれかであるかを決定するために用いるのではなく、候補の中からもっとも人狼らしいエージェントや村人陣営らしいエージェントを決定するために用います。

5. 更新倍率の学習

1 試合が終了後、更新倍率を試合結果から学習しています。試合中に倍率テーブルを更新した状況や条件を記憶しておき、試合後、その状況や条件が結果と矛盾していなければ更新倍率を大きくし、矛盾していれば更新倍率を小さくします。例えばある試合中にエージェント A がエージェント B を占った結果人狼だったと言ったとします。この場合、A が人狼、B が人狼というパターンに対応した倍率テーブルの値が非常に小さくなるような更新倍率となっています。しかしながら試合後、A、B とともに人狼であったという試合結果が得られた場合、この条件に対応した更新倍率の値は大きくなります。こうすることで、次の試合からは同じ条件の時に、A、B が共に人狼になるパターンの陣営推測テーブルの値は以前よりも少し大きくなり、推測の精度が上昇すると考えられます。

6. 更新条件と更新倍率

更新条件の詳細や更新倍率の初期値は `jingai table` パッケージの `LGJingaiTable.java` を参照してください。参考にさせていただいたチーム `cash` さんのものに少し条件を追加して、更新

倍率を少し変化させています.

■ 各役職アルゴリズム

その役職しか持っていないアルゴリズムを記載します. 狂人と人狼(と, 占い師も少し)はアルゴリズムをいくつか用意し, 学習を行った中で最適なものを利用する形にしています.

【村人】

人狼陣営の占い CO を減らす意図で, 低確率で占い CO して村人にスライドします
吊り回避のため, 黒出しされたら狩人 CO します

【霊能】

自分視点で矛盾があるプレイヤー(霊能や占い師)に投票します.

【狩人】

占→霊→村の順で護衛します. GJ 判定もします.

【占い師】

吊られそうでなく, 最も人狼の可能性が高いプレイヤーを占います.

黒出し先→偽占, 偽霊→グレーの順で投票します.

初日に CO するもの 2 つ(陣営推測テーブルの更新規則が異なる)と 1 日目に CO するもの 1 つ, 計 3 つのエージェントがあります.

【狂人の戦略】

- ・ 占い CO/役職 CO 黒出し狂人

役職 CO しているプレイヤーに黒出しし, 吊られそうなプレイヤーに投票します

- ・ 占い CO/狼同調狂人

最も人狼の可能性が高いプレイヤーに白出しし, そいつの Vote 宣言に投票します

- ・ 占い CO/霊能スライドあり狂人

占い CO 4 人以上で霊能にスライドします

占いなら人狼の可能性が低いプレイヤーに 3 連続黒出し, 霊能なら 0.5 の確率で黒出しします

自分の黒出しか最も人狼の可能性が低いプレイヤーに投票します

【人狼の戦略】

- ・ 役職 CO/役職噛み人狼

占いか霊能 CO, 日数の偶奇で噛み役職を決めます

- ・ 指定役職優先噛み人狼

占→狩→霊の順で優先噛みします

- ・ 霊能優先噛み人狼

噛みは霊能 CO を優先, vote は身内が占い CO したときの黒出し先に投票します

- ・ 狩人 CO 人狼

Vote 宣言を多数からされた場合狩人 CO します

