平成 29 年 8 月 27 日
cndl

第 3 回人狼知能大会 cndl チーム概要

1　プログラムの構成

本プログラムは、戦術等のアルゴリズムの実装に集中するため、本来の人狼知能フレームワークの上にイベント駆動型のフレームワークを搭載している。具体的には、1 日の開始や、会話、攻撃等のイベントが発生した時に各アルゴリズムやモデルが実装するイベントハンドラが呼ばれ、分散して情報を保持・更新する形式となっている。

プログラムの主たる要素は以下のとおり。

100 **試合内の累計情報・モデル（**metagame **パッケージ）**　各プレイヤーがそれまでに何勝しているか、役職推定用の累積発言行動情報等、試合をまたいだ情報を管理する。

試合状況モデル（model **パッケージ）**　現在の試合における各プレイヤーの行動や役職推定情報を管理する。

役職（role **パッケージ）**　自分が割り振られた役職において、どのタイミングでどの戦術を用いるかを定義する。

戦術（talk, target **パッケージ）**　ゲーム状態からどの行動（会話・投票）を選択して行うかを選択して返すクラス群。可能な範囲で小さい単位の実装とし、戦術の共有や差し替えの容易性を担保している。

フレームワーク（framework **パッケージ）**　上記の実装を統合し人狼知能フレームワークに接続する機構と、それまでの会話・投票履歴、役職別のカミングアウト済のプレイヤー、判明している役職等、判断を要さない基礎的な情報を管理する。

ユーティリティ群（tools **パッケージ）**　デバッグ用ログ出力メソッド、ゲームログの分類・整形、ゲームログを入力としてプレイヤーを動作させる等、開発・ログ解析に有用なユーティリティ群。

2　役職別の戦術

2.1　15 人村

15 人村の基本戦術は目立たないことである。これは、ESTIMATE 等の発言を実装した場合に、追放・襲撃率が上昇したためであり、また、その時点での推測情報を発言することよりも、自分が生き残って適切な投票をすることの方が自らの陣営の勝利に資することが予測されるためである。

村人　最初に最も EvilScore（後述）の高いプレイヤーに投票宣言を行い、1 ターン目以降は場の空気を少し読んで投票が集まっている人狼らしいプレイヤーに投票宣言。
投票は宣言通りに行う（ただし自分が追放されそうな場合には EvilScore を無視して生き残るために投票）。
村人陣営の投票宣言・投票は全て同じ。

占い師　初日 0 ターン目にカミングアウトし占い結果を伝える。2 日目以降は 0 ターン目で占い結果を伝える。占いで狼を発見できている場合にはそのまま結果を伝えるが、狼を発見できなかった場合、現在最も EvilScore が高いプレイヤーに黒出しをする。
0 日目の占い対象は、勝率の高いプレイヤー。1 日目以降は、EvilScore の高い方から順に占っていく（ただし占いと霊能は原則対象外）。

霊媒師　初日 0 ターン目にカミングアウト。 2 日目以降は 0 ターン目で霊媒結果を伝える。

狩人　SeerScore（後述）の高い生存プレイヤーを護衛。
4 日目以降、護衛対象の EvilScore が平均以上の場合（占い騙りの狼等）は、対象を変更する。

人狼　日中の会話では村人のフリをする。
投票においては、自分の票で仲間の狼の得票数が最多で無くなる場合には投票先を変更するが、基本的には大衆の意見に従って投票する。
裏切り者以外で、人狼陣営へのこれまでの投票数の多いプレイヤーを優先して襲撃。

裏切り者　日中の会話では占い師のフリをする。
初日占い師カミングアウトが自分以外に一人なら、その占い師に黒出し、そうでなければ狼っぽく無いプレイヤーに適当に黒出し。
3 日目までは狼っぽく無いプレイヤーに黒出しし、自分が黒出ししたプレイヤーに投票宣言。
4 日目以降はそれまでの発言は無視し、EvilScore の低いプレイヤーに投票。

2.2　5 人村

15 人村と同様に目立たないことを基本方針としており、使用するトピックは VOTE, COMINGOUT, DIVINED の 3 種のみである。これは、予備予選のログを分析したところ、これら以外の

発話が行われることは稀であり、また 3 種以外を使用すると生存率・勝率が芳しくないためである。

村人　初日 0 ターン目に、勝率トップのプレイヤーへ VOTE 宣言をする。これ以降は占い宣言を聞いた後、EvilScore トップのプレイヤーが更新された場合にそのプレイヤーに VOTE 宣言を行う。

最終的に EvilScore トップのプレイヤーへ投票する。

2 日目 0 ターン目に EvilScore トップへ投票宣言。初日と同様にスコア更新があれば Vote 宣言を改める。

裏切り者カミングアウトをしたプレイヤーがいればそれを避け、EvilScore のトップへ投票する。

占い師　0 日目の占い対象は、勝率の高いプレイヤー。

初日 0 ターン目にカミングアウトし、1 ターン目に占い結果を伝える。この際、占い判定が黒だった場合は正直に伝えるが、 判定が白だった場合は、EvilScore の最も高いプレイヤーを狼であったとして嘘の DIVINED 宣言をする。

意外に勝率が高い。狼を当てられればそれでよく、外れても裏切者と判断されるため、生存率が高いと考えられる。

二日目はほぼ確実に狼であるプレイヤーが分かる。そのプレイヤーが狼であると Divined によって伝え、投票を行う。

人狼　村人のフリをする。

自分以外で最も投票宣言の対象として票を稼いでいるプレイヤーに投票宣言・投票。

生存している自称占い師が 2 人であれば、どちらも残して素の村人を襲撃する。1 人である場合、RoleScore を用いて裏切者がどちらかを推測し、生存者が占い師であると判断した場合は襲撃する。そうでない場合は最も村人らしいプレイヤーを襲撃。

2 日目に裏切り者カミングアウトがあれば、自分は人狼カミングアウトをする。

裏切り者　1 日目 0 ターン目に占い師カミングアウト。自分以外の占い師カミングアウトが 1 人いればそのプレイヤーに、さもなければ EvilScore の最も低いプレイヤーに黒出し。

自分が黒判定を出した相手に投票する。自分以外の占い師カミングアウトは真占いである可能性が非常に高く、真占いを追放できればよし、できなくとも狼に裏切者であると伝えることができる。

自身に票を集めることを意図して、最終日に人狼カミングアウト。自身は対抗占いが黒判定を出した相手には投票せず、対抗占いが白判定を出した相手に投票する。情報がなければ EvilScore の低い方に投票する。人狼カミングアウトしたプレイヤーには投票しない。

3　役職推定アルゴリズム

3.1　行動頻度による推定

3.1.1　プレイヤー別の役職確率の事前分布

各プレイヤー i について、ある役職 r である“確率”（事前分布）$P_i(r)$ を設定する。$P_i(r)$ は、最初に各役職に割り当てられる人数の逆数として設定された後、人狼による襲撃によって人狼で無いことが確定した場合や、占い、霊媒、護衛の結果等によって確定的に更新される。

3.1.2　単純ベイズ法による会話トピックからの役職推定: TalkFrequancyModel

各プレイヤーの発言は他のプレイヤーの配役とは関係無く、ただ自身の役職によってのみ決まるという（かなり非現実的な）仮定を置き、d 日目の第 t ターンにおいて、プレイヤー i が会話トピック T の発話を行なう確率を $P_{i,d,t}(T)$ 、 i の役職が r でありかつ T の発話を行なう同時確率を $P_{i,d,t}(T,r)$ とそれぞれ書く。ここで会話トピックとは、ContentBuilder で生成できる 13 種（ATTACK は除いている）の述語であり、投票先や占い結果などは考慮しない。当然に $\sum_r P_{i,d,t}(T,r) = P_{i,d,t}(T)$ が成り立つ。

i が役職 r であるときに、T の発話を行なう条件付確率（尤度）$P_{i,d,t}(T|r)$ を以下の関係式により定義する。

$$P_{i,d,t}(T|r)P_i(r) = P_{i,d,t}(T,r)$$

以上を用いると、i が T の発話を行なった場合に、 その役職が r である条件付確率（事後確率）$P_{i,d,t}(r|T)$ を以下のように表わせる（ベイズの定理）。

$$P_{i,d,t}(r|T) = \frac{P_{i,d,t}(T,r)}{P_{i,d,t}(T)} = \frac{P_{i,d,t}(T|r)P_i(r)}{P_{i,d,t}(T)} = \frac{P_{i,d,t}(T|r)P_i(r)}{\sum_{r'} P_{i,d,t}(T|r')P_i(r')}$$

各発言は、それ以前の発言とは統計的に独立であるという（非現実的な）仮定を置くと、発話トピックの時系列 $\{T\}$ による事後確率 $P_i(r|\{T\})$ を各発言での尤度と事前確率の積に分解できる。($Z_i(\{T\}) = \sum_r P_i(r) \prod_{d,t} P_{i,d,t}(T|r)$ は規格化定数)

$$P_i(r|\{T\}) = \frac{1}{Z_i(\{T\})} P_i(r) \prod_{d,t} P_{i,d,t}(T|r)$$

上式により、一連の発話トピックからプレイヤーの役職を推定できる。

$P_{i,d,t}(T|r)$ は、過去の試合ログの発言頻度から計算したプレイヤーに依らない値を初期値として用い、1 試合が終了し各プレイヤーの役職が明らかになる度に、その試合での発話に基づいてプレイヤーごとに更新する。

3.1.3　リウェイト

上記の単純ベイズ法による推定では、その村における役職の人数は保存されない。すなわち例えば、襲撃・追放されたプレイヤー及び自分自身も含めた役職別事後確率の和 $S_{\text{人狼}} =$

$\sum_i P_i(\text{人狼}|\{T\})$ は、3 になるとは限らない（15 人村の場合)。 このため、その村における役職 r の総数を N_r として、

$$P'_i(r|\{T\}) = \frac{1}{Z'_i(\{T\})} \frac{N_r}{S_r} P_i(r|\{T\})$$

という補正を行っている（$Z'_i(\{T\})$ は規格化定数)。なお、これはあくまで補正であり、 補正後の $S'_r = \sum_i P'_i(r|\{T\})$ も N_r とは揃わない。

3.1.4 **行動頻度による推定**: ActFrequancyModel

単純ベイズ法による役職推定は、会話トピックだけでなくプレイヤーが行う任意の行動に拡張可能である。ただし、「プレイヤーの行動確率は自分の役職のみに依存して決まる」「ある行動はそれ以前の行動と統計的に独立である」という条件から大きく逸脱している場合、精度が下がると考えられる。

今回は、占い師の可能性が高い対象への投票、ESTIMATE 発言の際に黒出しをしたか白出しをしたかなど、対戦ログ等から村人陣営と狼陣営で違いが出ると考えられる 23 の行動について、会話トピックによる推定とは別に役職推定を行った。

3.2 ルールベースのスコアリング

行動頻度による推定が多少は補うものの、確率推定では行動の少ない（＝ステルス）プレイヤーに関する推測精度が下がる。そのため、通常の人狼ゲームと同様に占い師や霊能の情報も利用する必要がある。

3.2.1 **行動内容の論理矛盾スコアリング**: AgentReliablityModel

本大会ではほぼ全ての狂人が占い師カミングアウトを行うため、真占いとの弁別が必要である。確率モデルも利用するが、占いの発言が過去発言や状況、霊能結果と矛盾が無いかをチェックするルールベースのスコアリングをしている。

3.2.2 **占い結果の採用**: BelieveSeerModel

上記の AgentReliablityModel を用いて、最も信頼性の高い占い・霊能の結果を採用し、白黒の判定を行う。

3.3 各モデルの統合

3.3.1 RoleScore

TalkFrequancyModel 及び ActFrequancyModel の 2 つの方法で得られた事後分布を、プレイヤーごとに合計し、そのプレイヤーの役職スコアとして用いる。役職スコアが大きいほど、そのプレイヤーがその役職である可能性が大きいと考える。

3.3.2 EvilScore

人狼らしさを推定するスコアで、RoleScore を占い結果で大きく補正して得る。さらに、狼の占い師カミングアウトが少ないことが観測されているため、4 日目まではなるべく情報を得るために占い師・霊能者カミングアウトしたプレイヤーのスコアも補正している。生存している EvilScore 最大のプレイヤーが狼である確率は、進行と共に減少していくが、平均約 73% 程度である（予備本戦で自分が狼以外の役職だった 3968 試合での結果)。なお、ルールベースのアルゴリズムを外した場合には 77% まで上昇するが、前述の通りステルス系狼の発見率が下がることが想定されたため、最後まで残した。

3.3.3 SeerScore

RoleScore に、AgentReliablityModel のスコアを加味した占い推定用のスコア。精度は平均 51% 程度。