

人狼プログラム Thousand for CEDEC2016 ドキュメント

2016/8/21 大渡 勝己 (Katsuki Ohto)

実行方法

java 言語, プレーヤークラスは
jp.ac.u_tokyo.ecc.tanaka.ohto.aiwolf.ThousandPlayer
ただし他プログラムのソースコードをクラスパスを変えずに利用しているため、-cp で他プログラムを一緒に読み込まないよう注意。

コンセプト「メタ推定」

予備予選や予備決勝が開催されて、実際の対戦相手の棋譜を入手出来ることを利用し、決勝に出場する 15 プログラムの行動傾向を解析しどのエージェントが誰かを特定して行動決定に利用する。

基本戦略

第 1 回大会優勝の Udon プログラム (第 1 回 GAT 杯バージョン) をほぼそのまま利用しているため、各役職の戦略は Udon プログラムに準じます。

Thousand の独自の工夫として、Udon プログラムの行動決定の際の行動価値関数に、プログラムの特定による結果を加えます。具体的には、予備予選や予備決勝で高い勝率を残したプログラムにより投票、襲撃等しやすくなるようにします。特に予選で自分より上位だった Udon, cash, wasabi (敬称略) は優先度を大きくしています。

アルゴリズム

役職の推定等は Udon プログラムそのままなので、私に加えたエージェント特定アルゴリズムについて説明します。

①事前に得た棋譜から各プログラムの会話の内訳、各役職のカミングアウトのタイミングについて割合を計算

例) JuN1ro, Noppo プログラム予備決勝における Talk の内訳
(OVER, SKIP, VOTE, ESTIMATE のみ)

	OVER	SKIP	VOTE	ESTIMATE
JuN1ro	0 %	23 %	60 %	17 %
Noppo	64 %	5 %	28 %	3 %

②実際の試合で各エージェント番号（100 試合固定）ごとに会話の種類やカミングアウトのタイミングを集計（したはずだったが設計ミスがあり一部しか記録できていないようだ...）

③集計結果 α を割合に対するディリクレ分布として扱い、事前に調べた各プログラムの行動割合データ x を引数とする（対数）確率密度を計算

<資料> ディリクレ分布の確率密度関数

$$density(x) = \frac{\prod_i x_i^{\alpha_i - 1}}{B(\alpha)}$$

ここで $B(\alpha)$ は多変数版のベータ関数

$$B(\alpha) = \frac{\prod_i \Gamma(\alpha_i)}{\Gamma(\sum_i \alpha_i)}$$

ただし $\Gamma(z)$ はガンマ関数

$$\Gamma(z) = \int_0^{\infty} t^{z-1} e^{-t} dt$$

④計算した確率密度を正規化して、各エージェント番号のプログラムがどのプログラムであるかの確率予測テーブルを作る

例) 予備決勝 8 / 17 11:00 における実際の推定結果%(50 試合時点のころ)

真\推	Ud	ca	wa	Th	ca	ic	m_	Ju	No	Li	GO	TW	Te	pr	da
Udo	89	0	0	3	0	0	0	0	0	0	0	0	0	0	0
cas	0	99	0	0	3	0	0	0	0	0	0	2	0	0	0
was	0	0	65	1	0	0	0	0	0	0	0	0	16	0	0
Tho	0	0	0	79	0	0	0	0	0	0	0	0	0	0	0
car	0	0	0	0	0	61	0	0	0	0	27	16	0	0	0
ich	11	0	0	0	86	0	0	0	0	0	0	0	0	0	0
m_c	0	0	0	0	0	0	99	0	0	0	0	0	0	0	99
JuN	0	0	0	0	0	0	0	0	0	0	0	0	1	75	0
Nop	0	0	0	0	0	0	0	0	13	0	0	0	75	0	0
Litt	0	0	0	0	11	38	0	0	0	0	2	52	0	0	0
GOD	0	0	0	0	0	1	0	0	0	0	71	30	0	0	0
TWR	0	0	0	0	0	0	0	54	0	47	0	0	0	0	0
Tea	0	0	0	0	0	0	0	0	86	0	0	0	4	0	0
pre	0	0	35	17	0	0	0	0	0	0	0	0	1	25	0
dai	0	0	0	0	0	0	0	45	0	0	0	0	3	0	0

上位陣は特徴的なプログラムが多いので推定できていますが、バグや不具合が色々あり中位以下は全然合っていないですね...

人狼知能系使用ライブラリ

Thousand のソースコード群には以下のソースコードが含まれます。

Udon プログラム 2016 年 GAT 杯バージョン 中田洋平氏 Java

(予備予選、予選、予備決勝で使用、**決勝のベースプログラム**、C++移植しようとして断念)

kSugar プログラム 2016 年 GAT 杯バージョン kSugar 氏 Java

(予備予選、予備決勝で使用、C++移植しようとして断念)

wasabi プログラム 2016 年 GAT 杯バージョン 水越俊希氏 Java

(予備予選、予選、予備決勝で使用、C++移植しようとして断念)

AIWolfAgentLibrary 2016/8/8 時点 carlo 氏 Java

(試合結果の集計に利用しようとしてインスタンスを作ったが結局間に合わず思考には利用しなかった)

cash プログラム 2016 年 GAT 杯バージョン 原田慧氏 Python -> C++

(C++へ移植して対戦実験相手として利用、ベイズ系推定の実験)

calc_score.py バージョン不明 大槻恭士氏 Python

(細部を書き換えて勝率計算に利用)

上記プログラムの作者の皆様に深く感謝致します。

蛇足：予選戦略

実際の対戦相手（潜伏している者もいたが...）が予備予選に出場していることを利用し以下の戦略で予選を戦いました。

①kSugar, Udon, wasabi の 3 プログラム（GAT 杯上位の Java プログラム）のうち、予備予選の 1 試合ごとにランダムにどれかのプログラムを使用し、どれを使用したかを出力

②フィードバックされる棋譜と出力ログを照らし合わせて、各役職でどの勝率が高かったか調べる

③勝率が高かった組み合わせで固定して予選に参加

実際には予備予選の終盤で kSugar が落ちてしまったので Udon, wasabi の 2 つのみを利用し、以下の組み合わせとしました。

村人	占い師	霊媒師	狩人	狂人	人狼
Udon	Udon	wasabi	Udon	Udon	wasabi