

チーム名：TOT プログラマ：大槻恭士 所属：山形大学

昨年に引き続きGATの人狼知能部門に参加させていただきました。学習した確率分布をファイルから読み込む必要があったので、今回はJavaで作成しました。

アルゴリズム

基本的にはプラットフォーム付属のサンプルエージェント（これも私が作成）を簡略化したものなので、各役職のアルゴリズムについては省略します。

そしてGAT2017では、村人エージェントにおいてランダム選択の代わりに適応確率モデルに基づくエージェント選択を導入しました。

記号の定義

$P(B_k|R_j)$ ：役職が R_j であるとき平均エージェントが行動 B_k を取る確率（平均モデル）

$P(B_k|A_i, R_j)$ ： A_i の役職が R_j であるとき A_i が行動 B_k を取る確率（エージェントモデル）

適応確率モデルに基づくエージェント選択

行動 B_k が観測されたとき、行動の主体が役職 R_j のエージェント A_i である確率は

$$P(A_i, R_j | B_k) = \frac{P(A_i)P(R_j|A_i)P(B_k|A_i, R_j)}{P(B_k)}$$

ですが、人狼知能大会の場合 $P(A_i), P(B_k)$ は無視でき、 $P(R_j|A_i) = P(R_j)$ なので、 B_k が観測されたときの A_i が R_j である事象の対数尤度関数は、 $L_{B_k}(A_i, R_j) = \log_{10} P(R_j) + \log_{10} P(B_k|A_i, R_j)$ となります。さらに、対戦中に役職 R_j の A_i が行動 B_k をした回数 $n(A_i, R_j, B_k)$ と A_i が R_j になった回数 $n(A_i, R_j)$ より、 $P^*(B_k|A_i, R_j) = n(A_i, R_j, B_k)/n(A_i, R_j)$ のように短期エージェントモデルを計算します。そして、平均モデルと短期エージェントモデルの重み付き混合で各エージェントに適応したモデルを求めます。この場合対数尤度関数は次式のようになります。

$$\hat{L}_{B_k}(A_i, R_j) = \log_{10} P(R_j) + (1-w) \log_{10} P^*(B_k|A_i, R_j) + w \log_{10} P(B_k|R_j)$$

提出したエージェントでは、村人エージェントが複数のエージェントから投票先を選ぶ場面で、従来のランダム選択ではなく人狼である尤度が高いエージェントに投票するようにしました。この場合役職が人狼固定なので上式右辺第1項は不要になり、 $(1-w) \log_{10} P^*(B_k|A_i, WEREWOLF) + w \log_{10} P(B_k|WEREWOLF)$ が最大となるエージェントを選ぶことになります。

今回は w を振って最適値を決定する時間がなく、 $w = 1.0$ で提出しました。つまり適応はしていないことになります。なお、平均モデルはサンプルエージェント14体と1万回対戦して求めました。

行動とその確率

今回は学習回数が100回と少なく、さらに勝率を上げるには早い段階で推定精度を上げる必要があるため、「占い師カミングアウト」、「霊媒師カミングアウト」、「村人に投票」、「占い師に投票」、「霊媒師に投票」の5行動としました。

また、行動の系列の確率は学習データの少なさを考慮してunigramモデルで計算しています。つまり行動の順番は考慮せず、単に同時確率（確率の積≡尤度和）で計算しています。