F-054

不完全情報ゲームにおける複数エージェントの協調活動のための状態推定
State Estimation to Enable Multi-Agent Cooperation in Imperfect Information Games

久保田　晴明†　関根　栄子†
Seimei Kubota　Eiko Sekine

1．はじめに

エージェントの協調活動により特定の問題を解決するシステムが注目を集めている[1][2]．そこにおいて，個々のエージェント間の連携を図るリーダーエージェントを設定して各エージェントの持つ情報や環境情報をすべて把握させるか，エージェント間で情報を共有することにより協調活動を行うなど，協調活動を行う際の不確定性が小さくなるように問題設定がされていることが多い．しかし，麻雀に代表される不完全情報ゲームにおいては，エージェントごとに持っている情報が異なるうえゲーム進行における不確定性が大きいため，エージェント同士がお互いの状態を把握して直接的に協調活動を行うことは困難である．

本研究では，不完全情報ゲームにおいて，複数のエージェントの協調活動により 1 人のプレイヤーを強化するシステムの開発を目的とする．具体的には，麻雀を子供向けに簡略化したゲームである「ドンジャラ」を対象とし，ゲームを行う中でトレーニング対象となるプレイヤーの能力を向上させるようなエージェントの行動を強化学習により学習させる．不完全情報ゲームにおいて直接的な協調活動を行うことが難しいという問題を解決するために，手牌推定と強化学習を組み合わせた協調型トレーニングシステムを提案する．

2．問題設定

ドンジャラは麻雀を子供向けに簡略化した 2～4 人で遊ぶボードゲームである．各プレイヤーは他のプレイヤーには見えない 8 枚の牌を持ち，手番ごとに牌山から 1 枚引き，合計 9 枚の手牌から 1 枚を捨てるという行為を繰り返し，先に特定の役を揃えたプレイヤーが得点を得る．これを複数ラウンド繰り返し，最終的に持ち点の多いプレイヤーが勝者となる．図 1 で示すように，ドンジャラにおいて公開情報となるのは，各プレイヤーの現在得点，

図 1　ドンジャラの公開情報と非公開情報

†茨城大学, Ibaraki University

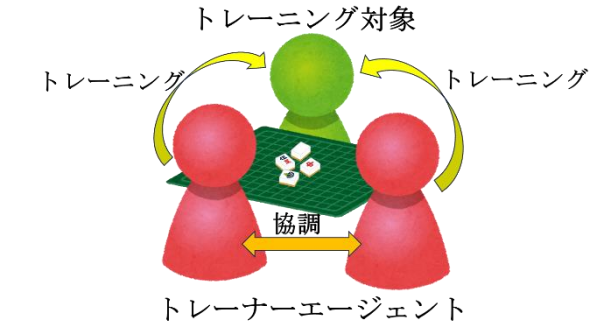


図 2 提案システムのイメージ

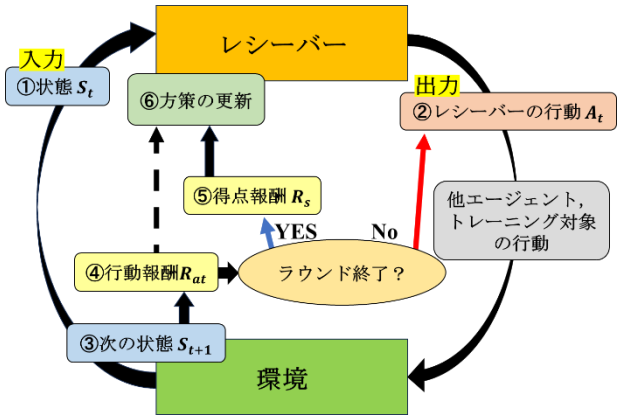


図 3　トレーナーエージェントの強化学習の流れ

自分の手牌，各プレイヤーが現在のラウンドで捨てた牌である．他のプレイヤーの手牌と牌山の牌の順番は非公開情報となる．本研究では，ドンジャラを基に本研究用にアレンジしたゲームをトイプロブレムとして用いる．ドンジャラとトイプロブレムの相違点として，使用する牌は 9 種類の牌を各 9 枚ずつで合計 81 枚のみとし，鳴きは実装せず，リーチはあと 1 枚で上がれる状態になった際に自動で宣言されるものとした．また，役は，9 枚すべて同じ牌を揃える役，同じ牌 6 枚と別の種類の同じ牌 3 枚を揃える役，3 枚の同じ牌を 3 種類揃える役の 3 つを設定した．

3．システム構成

3.1 強化学習の流れ

提案システムでは，図 2 で示すように，2 体のトレーナーエージェントと 1 人のトレーニング対象の計 3 人のプレイヤーがドンジャラをプレイするという状況を想定する．2 体のトレーナーエージェントが協調活動によりトレーニング対象の能力を向上させる行動を強化学習により学習する．図 3 に，トレーナーエージェントの強化学習の流れを示す．2 体のトレーナーエージェントの報酬設計は同一であり，ここでは，1 体のトレーナーエージェントの

強化学習の流れを説明する．対象となるトレーナーエージェントを便宜上レシーバーと呼称する．状態 S_t （$t \geq 1$）はレシーバーから見えている情報を表し，現在のラウンド数，親，各プレイヤーの得点，各プレイヤーのリーチの有無，自分の手牌，各対戦相手の手牌，各プレイヤーの捨て牌で構成される．ここで，各対戦相手の手牌は直接的に観測できないので，Decision Transformer[3]を用いた手牌推定を行う．行動 A_t はレシーバーが自分の手番に捨てる牌であり，牌の種類に対応した 0 から 8 の数値をとる．レシーバーは自分の手番が来る度に，状態 S_t を入力として行動 A_t を出力する．レシーバーの行動のトレーニング対象に対する影響により行動報酬 R_{at} が与えられる．ラウンド終了時，行動報酬 R_{at} と 1 ラウンドの結果に応じて与えられる得点報酬 R_s を用いてレシーバーの方策を更新する．

3.2 報酬設計

レシーバーに与えられる報酬 R_t は行動報酬 R_{at} と得点報酬 R_s により次式で定義される．

$$R_t = wR_{at} + (1-w)R_s \tag{1}$$

ここで，$0 < w < 1$ は重みである．

行動報酬 R_{at} はつぎで定義する．

$$R_{at} = r_b(N, A) \times m \tag{2}$$

$r_b(N, A)$（> 0）はレシーバーの行動に対するトレーニング対象への影響度であり，ここでは基本報酬と称し，つぎで定義する．

$$r_b(N, A) = 1 + \varepsilon - N/|A| \tag{3}$$

ここで N は，レシーバーが実際の行動とは異なる行動をとった場合をシミュレーションし，そのシミュレーションにおいてレシーバーの各行動に対してトレーニング対象がとった行動が，トレーニング対象の実際の行動と同じになる数を表している．$|A|$ はレシーバーがとり得る行動の候補数を表す．したがって，$N = |A|$ となるのは，レシーバーがどんな行動をとったとしてもトレーニング対象への影響がないときである．$\varepsilon > 0$ は，$r_b(N, A)$ が 0 になることを避けるために導入した．図 4 に基本報酬の計算例を示す．図 4 の下段はレシーバーの手牌を表し，上段はレシーバーの捨てた牌を受けてトレーニング対象が捨てた牌の例を表している．このとき，レシーバーがとり得る行動の候補は 0，1，4，5，8 のいずれかの牌を捨てることであるため，$|A|$ は 5 となる．レシーバーが実際に捨てた牌が 0 のとき，トレーニング対象が捨てた牌は 1 であり，レシーバーが 1，5 を捨てた場合でもトレーニング対象が 1 を捨てるため，N は 3 となる． m はレシーバーの行動の良し悪しにより基本報酬を補正する係数であり，ここでは報酬倍率と呼称する．報酬倍率 m は，シャンテン数 γ（あと最低何手で上がれるかを示す数）と振り込み（自分が捨てた牌により他のプレイヤーが上がること）の有無によりつぎで決まるものとした．

$$m = \begin{cases} M(C_a) & (\gamma \text{ が減る}) \\ 1 & (\gamma \text{ が変わらない}) \\ -M(C_a) & (\gamma \text{ が増える or 振り込み有}) \end{cases} \tag{4}$$

ここで M はレシーバーの行動回数 C_a が増えるほど大きくなる関数でありつぎで定義する．

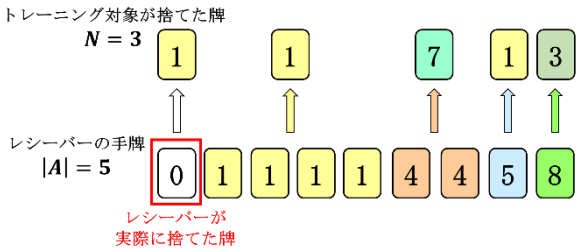


図 4 基本報酬の計算例

$$M(C_a) = \alpha^{C_a} \tag{5}$$

ここで $\alpha > 1$ は行動報酬の範囲を調整するための係数である．レシーバーがシャンテン数の減る行動をすると，レシーバーが上がりに一手近づくので，トレーニング対象にとって，それ以降のレシーバーの行動がより脅威となり，このラウンドで上がる，または他のプレイヤーに振り込まないためにとるべき行動を考えるようになる．したがって，シャンテン数が減る行動はトレーニング対象にいい影響を与える（$m > 1$）．逆に，レシーバーが他のプレイヤーに振り込むなどの利敵行為はトレーニング対象が能力を向上させるうえで妨げとなるためトレーニング対象に悪い影響を与える（$m < 0$）．これにより，行動報酬はレシーバーがトレーニング対象に対していい影響を与えた際（$m > 1$）に報酬が大きくなり，悪い影響を与えた場合（$m < 0$）は負の報酬となる．行動報酬を最大化することによりトレーニング対象の能力の向上が期待される．

得点報酬 R_s はつぎで定義する．

$$R_s = s_g \times r_h \times W(C_a) \tag{6}$$

s_g は，レシーバー自身が上がった場合は 1，他のプレイヤーが上がりかつレシーバーがそのプレイヤーに振り込んでいる場合は-1，それ以外の場合は 0 をとる．r_h は揃えた役に応じた得点 h により値が決まる数で，ここでは役報酬と呼称し，つぎで定義する．

$$r_h = \delta h \tag{7}$$

ここで $0 < \delta < 1$ は h の桁数に応じて決まる係数である．関数 W は上がるまでの行動回数 C_a が少ないほど大きくなる関数であり，ここでは上がり関数と呼称し，つぎで定義する．

$$W(C_a) = -\beta {C_a}^2 \tag{8}$$

ここで $\beta > 0$ は得点報酬の範囲を調整するための係数である．レシーバーが上がったゲームに対し得点報酬を与えることで，レシーバーがトレーニング対象にいい影響を与える行動をとりやすくすることができる．

3.3 方策の更新

本研究では，方策の更新に反事実的後悔最小化（MCCFR）[4]を用いる．MCCFR は，1 回の行動で得られた報酬とシミュレーションにより得られた報酬の最大値との差を後悔として累積し，それを最小化するように方策を更新するアルゴリズムである．2023 年に登場しオンライン麻雀ゲーム「天鳳」においてわずか 1 か月で 10 段

に到達した麻雀 AI「Lucky J」[5]が用いており，その有用性が示されている．

４．手牌推定

4.1 手牌推定の概要

手牌推定はトレーナーエージェントが協調活動を行う際に，対戦相手の手牌がもつ不確定性を緩和し，トレーナーエージェント同士が間接的に協調活動をするために行う．麻雀の手牌推定には Transformer を用いた教師あり学習による手法が提案されているが[6]，本研究ではドンジャラを基にしたオリジナルの問題設定を用いる都合で教師データの用意が難しいため，強化学習で手牌推定の学習を行う．この手牌推定には，強化学習向けに設計された Transformer である Decision Transformer を用いる．Decision Transformer では，過去から現在までの各ステップにおける「累積報酬和 $\hat{R}_t$，状態 s_t，行動 a_t」を式(9)のような軌跡表現 τ にしたものを入力とし，現在のステップにおける行動 a_t を出力する．

$$\tau = \left(\hat{R}_1, s_1, a_1, \hat{R}_2, s_2, a_2, \cdots, \hat{R}_t, s_t, a_t,\right) \tag{9}$$

手牌推定ではステップ数が大きいほど推定の精度が上がることが望ましいため，過去のデータを入力とする Decision Transformer は手牌推定の問題において有用であると考えられる．

4.2 手牌推定の状態・行動・報酬

手牌推定において，図 5 に示すように，盤面から得られる情報（現在のラウンド数，親，各プレイヤーの得点，各プレイヤーのリーチの有無，推定を行う主体の手牌，各プレイヤーの捨て牌）を系列化したものを状態 s_t とし，推定した各対戦相手の手牌を下家，上家の順に系列化したものを行動 a_t とする．ここで，下家は自分から見て次のプレイヤー，上家は自分から見て前のプレイヤーを表す．手牌推定の報酬R_{dt}をつぎのように設計する．

$$R_{dt} = \left(w_d {c_n}^2 + (1 - w_d){c_p}^2 - penalty\right) \times \left(1 + 0.05 \times (t - 1)\right) \tag{10}$$

ここでc_n, c_p はそれぞれ，下家と上家の手牌の一致率（推定した手牌と実際の手牌が共通している枚数の割合）であり，$0 < w_d < 1$ は重みである． $penalty$ は，牌の種類ごとに，枚数上限である 9 枚を超えるというありえない推定を行った場合にペナルティ定数 p を，そうでないなら 0 をとる値である．尚，一度の推定で複数の牌に対してペナルティが発生した場合でもペナルティは 1 回しか数えないものとする．ゲーム後半の推定をより重要視するため，ステップが進むにつれて報酬の範囲が大きくなるように設定する．この報酬設計により，ゲームが進むにつれて手牌推定の推定精度が向上することが期待できる．

4.3 手牌推定モデル評価

提案した手牌推定モデルの有効性を確認するための評価実験を行った．ある 1 ラウンドのゲーム用意し，ペナルティ定数 p を 1.0 と 2.0 の 2 種類としてそれぞれ学習させる．学習したモデルの下家と上家の手牌の一致率をランダム手法と比較した．ここで，ランダム手法とは，推定する主体から見えていない牌の中からランダムに出力を決定する手法である．この実験で用いたパラメータを表 1 に示す．

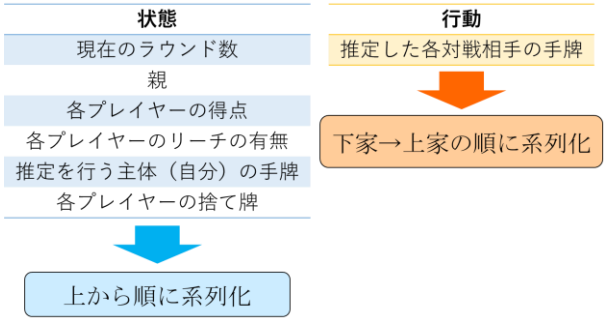


図 5　手牌推定における状態と行動

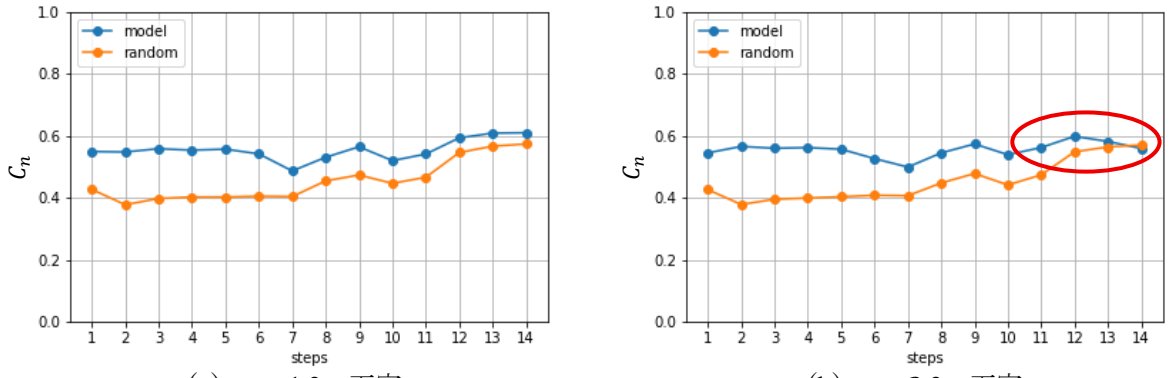
表 1　提案モデルのパラメータ

パラメータ	設定値
主要ライブラリ	PyTorch 2.5.1 + cuda 11.8 Transformers 4.45.2
重み $\boldsymbol{w_d}$	0.5
学習率	10^{-4}
状態次元数	44
行動次元数	18
隠れ層次元数	256
活性化関数	ソフトマックス関数
最適化関数	Adam
方策更新手法	REINFORCE
割引率	0.8

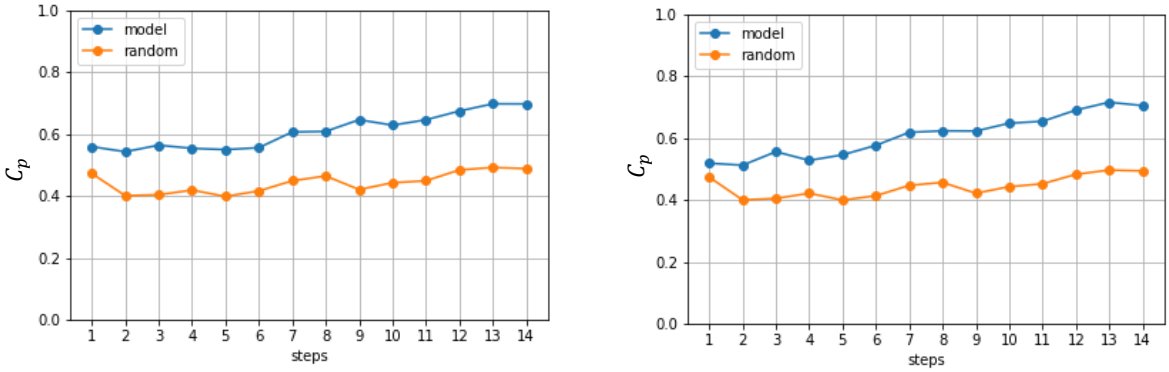
学習回数は 100.000 回とした．実験結果を図 6 に示す．一致率の評価回数は 3000 回として，その平均を示した．グラフの横軸はステップ数，縦軸は手牌の一致率を示しており，青い値が提案モデルの手牌の一致率，オレンジの値がランダム手法の手牌の一致率を示している．ペナルティ定数を 1.0 と 2.0 に設定した場合の結果を示した．ランダム手法では，ステップが進むにつれて一致率が上昇している傾向があるが，これはステップが進むにつれてランダム手法により選ばれる牌の選択肢が少なくなるためである．提案モデルでも同様の傾向がみられるが，特に，上家では一致率の向上が顕著に見てとれる．ペナルティ定数での比較では，$p = 2.0$ のときに下家のゲーム終盤の一致率がランダム手法を下回っているため，ペナルティ定数は 1.0 のほうが適当であると考えられる．これらから，提案した手牌推定モデルの報酬設計は適切であると考えられる．

５．トレーニング効果の評価

トレーニング効果の評価は，トレーニング対象となるプレイヤーと 2 体の仮想プレイヤーを対戦させることにより行う．仮想プレイヤーは，プレイ方針に応じた行動の確率分布に従って行動を選択するのみであり，トレーニング対象の能力を向上させるような行動は行わない．仮想プレイヤーのプレイ方針は，より少ない手数での上がりを目指す「速度重視型」，より高い得点の獲得を目指す「得点重視型」，他のプレイヤーに振り込まないようにプレイする「守備重視型」，状況に応じて臨機応変に打ち方を変える「バランス型」の 4 種類を設定する．トレーニング効果の評価に用いる指標を表 2 に示す．こ



（a）$p = 1.0$，下家　　（b）$p = 2.0$，下家



（c）$p = 1.0$，上家　　（d）$p = 2.0$，上家

図 6　手牌の一致率の比較

の指標は，麻雀においてプレイヤーの能力を評価する際に一般的に用いられている指標である．対戦によって得られたトレーニング対象と仮想プレイヤーの指標を比較し，トレーニング対象の向上した能力の傾向について評価を行う．

表 2　トレーニング効果の評価指標

評価指標	説明
和了率	1 ゲームあたりの 上がった割合
放銃率	1 ゲームあたりの 振り込んだ割合
平均和了点	上がったゲームあたりの 平均得点
リーチ率	1 ゲームあたりの リーチを宣言した割合
シャンテン数の変化率	1 ゲーム内の シャンテン数の推移

6．まとめ

不完全情報ゲームにおいて，手牌推定と強化学習を組み合わせた協調型トレーニングシステムを提案した．Decision Transformer を用いた手牌推定モデルを提案し，手牌推定の報酬設計が適切であることを確認した．この手牌推定モデルを用いた協調型トレーニングシステムのトレーニング効果は発表時に報告する．

謝辞

本研究は統計数理研究所共同利用登録（2025-ISMCR-P-0009）の助成を受けたものです．

参考文献

[1] 石川翔太，荒井幸代：渋滞低減に向けた路車間・車車間協調を実現する自動運転方策の学習法，人工知能学会論文誌，Vol.34 ， No.1， D-I55_1-9 (2019)
[2] 進化型オートメーションの実現に向けたロボット自律協調技術の開発，https://www.hitachihyoron.com/jp/archive/2010s/2019/03/05b05/index.html ， 2025 年 6 月閲覧
[3] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, Igor Mordatch : Decision Transformer: Reinforcement Learning via Sequence Modeling, arXiv:2106.01345 (2021)
[4] Noam Brown, Tuomas Sandholm：Superhuman AI for multiplayer poker， Science, 365, pp.885-890 (2019)
[5] 腾讯：腾讯 AI 登顶国际麻将平台，刷新全球最好成绩，https://mp.weixin.qq.com/s/yGH8rH05XvSFqWMl4VZWFw ，2025 年 6 月閲覧
[6] 大神卓也，奈良亮耶，天野克敏，今宿祐希，鶴岡慶雅：Transformer を用いた麻雀における手牌推定，ゲームプログラミングワークショップ 2022 論文集，2022 巻，pp.151-158(2022)
[7] Junjie Li， Sotetsu Koyamada， Qiwei Ye， Guoqing Liu， Chao Wang， Ruihan Yang， Li Zhao， Tao Qin， Tie-Yan Liu， Hsiao-Wuen Hon : Suphx: Mastering Mahjong with Deep Reinforcement Learning, arXiv:2003:13590 (2020)
[8] DOWANGO MEDIA VILLAGE：深層学習麻雀 AI「NAGA」，https://dmv.nico/ja/articles/mahjong_ai_naga/ ，2025 年 6 月閲覧