The 32nd Annual Conference of the Japanese Society for Artificial Intelligence, 2018

Neural Fictitious Self-Play における探索由来のデータを含めない教師あり学習による性能改善

Excluding the Data with Exploration from Supervised Learning Improves Neural Fictitious Self-Play

河村 圭悟 *1　鈴木 潤 *2*3　鶴岡 慶雅 *4
Keigo Kawamura　Jun Suzuki　Yoshimasa Tsuruoka

*1東京大学大学院工学系研究科
Graduate School of Engineering, The University of Tokyo

*2NTT コミュニケーション科学基礎研究所
NTT Communication Science Laboratories, NTT Corporation

*3理化学研究所 革新知能統合研究センター
RIKEN Center for Advanced Intelligence Project

*4東京大学大学院情報理工学系研究科
Graduate School of Information Science and Technology, The University of Tokyo

Neural fictitious self-play (NFSP) is a method for solving imperfect information games. While methods developed in recent years such as counterfactual regret minimization or DeepStack require the state transition rules of the games, NFSP works without them. In this paper, we propose to exclude the exploration data from the supervised learning component in NFSP and keep the probability of exploration, in order to explore without breaking the average strategy. We show that this change significantly improves the performance of NFSP in a simplified poker game, Leduc Hold'em, and compare the results for different exploration probabilities.

1. はじめに

人工知能分野の研究対象として，ゲーム AI が盛んに用いられている．その理由の一つは，実世界の問題に初めから人工知能の技術を適用しても実世界の問題が複雑すぎて技術の到達点（マイルストーン）を評価するのが困難となるため，行動の制約や報酬などのルールが厳密に定められたゲームにおいて技術を確立し，徐々に複雑なゲーム設定へと対象を移行することで，技術を効果的に高めていくためである．従って，現実の問題設定により近い不完全情報ゲームを対象とし，高い性能を発揮するプレイヤを作ることができれば，実問題への人工知能技術の応用の可能性を大きく進展できると考えられる．

不完全情報ゲームの文脈では，Texas Hold'em というポーカーゲームがよく題材として用いられる．Bowling らは，Texas Hold'em のベット額を制限した heads-up limit Texas Hold'em (HULHE) の解を，counterfactual regret minimization+ (CFR+) [Tammelin 14] という手法を用いて求めることに成功した (essentially weakly solved) [Bowling 15]．また，Moravčík らは，相手の手を仮定した時の最終的な期待報酬をニューラルネットワークで予測して CFR+を用いる DeepStack [Moravčík 17] というアルゴリズムを提案し，HULHE より複雑なゲームである heads-up no-limit Texas Hold'em (HUNL) においてプロのポーカープレイヤに勝利した．ほぼ同時期に，Brown らは，抽象化したゲームを探索して相手の手に応じて部分ゲームを生成し CFR+を用いて解くことで戦略を得る，nested subgame solving と呼ばれるアルゴリズムを提案し，このアルゴリズムを用いた AI である Libratus は HUNL においてプロのポーカープレイヤに勝利した [Brown 17]．

これらのアルゴリズムは，いずれもゲームの木を探索することで戦略を求める手法である．したがって，これらの手法を適用するには，agent があらかじめゲームの内容や状態遷移規則を全て知っておく必要がある．しかしながら，実世界の問題を解く場合には，環境の状態遷移規則が初めからわかっているという状況は考えにくく，未知の環境と相互作用しながら agent が状態遷移規則を把握していくような設定で問題を解くことができる手法が望まれる [河村 17]．

Heinrich らは，不完全情報ゲームで古くから用いられてきた Fictitious Play (FP) [Brown 51] と呼ばれる手法に強化学習の手法を応用して，ゲームの状態遷移規則を知ることなくゲームの解を求める Neural Fictitious Self-Play (NFSP) と呼ばれる手法を提案した [Heinrich 16]．この手法は，従来標準型ゲーム (Normal Form Game) で表現されたゲームにしか用いることができなかった FP を，展開型ゲーム (Extensive Form Game) で表現されたゲームにも適用できるようにした Fictitious Self-Play (FSP) と呼ばれる手法 [Heinrich 15] に，ニューラルネットワークを用いた教師あり学習と，ニューラルネットワークを Q 学習に応用した Deep Q-Network (DQN) [Mnih 15] を用いたものである．NFSP は，HULHE において事前の抽象化や簡略化といった前提知識を用いることなく既存のプレイヤに匹敵する性能が得られることが知られている．

本研究では，NFSP の教師あり学習が最適応答戦略の平均戦略を計算することを意図していることに着目し，教師あり学習の学習データに探索で得たデータを含めないようにすることで，NFSP の性能を向上させることができることを実験的に示す．

2. 関連研究

2.1 標準型ゲームと展開型ゲーム

標準型ゲームと展開型ゲームは，どちらも不完全情報ゲームを記述するモデルである．

展開型ゲームは木構造に基づく表現方式で，木のノードにあたる状態 s，各状態 s に対するターンプレイヤ $p \in P \cup \{c\}$，木のエッジにあたる行動 a がある．ここで，P はゲームプレイヤの集合を表しており，プレイヤのターンではそのプレイヤが行動の意思決定を行う．また，c は偶然手番を表しており，

連絡先: 河村圭悟，東京大学大学院工学系研究科電気系工学専攻，kkawamura@logos.t.u-tokyo.ac.jp

偶然手番では行動は各状態に対して事前に定まっている確率分布 $f(s)$ に従って生起する．木のリーフにあたる終端状態 z にゲームが到達すると，各プレイヤは報酬 $r_p(z)$ を得る．不完全情報ゲームでは，同じ状態であってもプレイヤによって観測できる情報が異なる．プレイヤ p から見て区別できないノードの集合を情報集合 I_p と呼ぶ．

展開型ゲームにおけるプレイヤ p の振る舞いは，p がターンプレイヤであるような任意の情報集合 I_p に対して行動集合 A 上の確率分布を与える関数 $\sigma_p(I_p)$ によって定まる．この関数を**戦略** と呼ぶ．各プレイヤの戦略の組み合わせ $\sigma = \sigma_1, \cdots, \sigma_N$ は戦略プロファイルと呼ばれる．プレイヤ p の期待報酬は戦略プロファイルに依存し， $r_p(\sigma)$ で表される．プレイヤ p 以外の戦略 σ_{-p} を固定したとき，期待報酬を最大化するような戦略 $\underset{\sigma_p}{\operatorname{argmax}}\, r_p(\sigma_p, \sigma_{-p})$ を σ_{-p} に対する **最適応答戦略** と呼び， $b_p(\sigma_{-p})$ で表す．どのプレイヤも，自身の戦略が自身以外の戦略に対する最適応答戦略になっているとき，その戦略プロファイルは **ナッシュ均衡戦略** であるという．

一方，標準型ゲームではこのような木構造を用いずにゲームを表現する．標準型ゲームでは，プレイヤ $p \in P$ はいくつかの**純粋戦略** a_p^N を持っており，各プレイヤが同時にどれか 1 つの純粋戦略を選択することでゲームが行われ，選択された純粋戦略の組み合わせに応じて各プレイヤは報酬 $r_p^N\left(\left\{ a_{p'}^N \mid p' \in P^N \right\}\right)$ を得る．各純粋戦略は，展開型ゲームにおいて戦略を決定的な (いずれかの行動のみが確率 1 で選択されるような) 確率分布にしたものに相当する．標準型ゲームでは，純粋戦略上の確率分布を**混合戦略**と呼び， σ_p^N で表す．標準型ゲームにおいても，展開型ゲームと同様に最適応答戦略，ナッシュ均衡戦略などを定義することができる．

2.2 Neural Fictitious Self-Play

FP は，標準型ゲームのナッシュ均衡戦略を求めるアルゴリズムである．FP では，最適応答戦略を用いて，混合戦略を

$$\sigma_{p,t+1}^N \leftarrow \frac{t}{t+1}\,\sigma_{p,t}^N + \frac{1}{t+1}\,b_p^N\left(\sigma_{-p,t}^N\right) \qquad (1)$$

と更新する．すなわち，各時刻の最適応答戦略を平均するように混合戦略を更新する．このアルゴリズムは，2 人零和ゲームなどいくつかの条件でナッシュ均衡戦略に収束することが示されている．

FSP は，FP を展開型ゲームでも適用できるようにし，さらに関数近似を用いて大きなゲームにも適用できるようにしたアルゴリズムである．標準型ゲームにおける混合戦略 σ_p^N と展開型ゲームにおける戦略 σ_p は，表現方法は異なるものの，不完全情報ゲームにおけるプレイヤの振る舞いを表しているという点では同じである．そこで，混合戦略 σ_p^N と戦略 σ_p が同じ振る舞いを示しているとき，σ_p^N と σ_p を実現等価であるという．ここで，戦略 σ_1 と σ_2 が同じ振る舞いを示すとは，任意の状態 s に対して，他のプレイヤ $-p$ が s に到達するように行動を選択した場合に状態 s に到達する確率 $x(s)$ が，プレイヤ p が σ_1 に従った場合と σ_2 に従った場合で等しいことを指す．この概念を用いて，FP のプロセスを展開型ゲームでの実現等価なプロセスに置き換えたのが Extensive-form Fictitious Play (XFP) [Heinrich 15] である．FSP ではさらに，XFP のプロセスが最適応答戦略を求めることと平均戦略を求めることの 2 つに分けられることに着目し，最適応答戦略を求める部分を強化学習で，平均戦略を求める部分を教師あり学習で近似している．これによって，XFP におけるすべての情報集合における確率分布を保持・更新しなければならないという問題点を解消し，かつ最新の機械学習のアルゴリズムを適用することができるようになっている．

FSP は特定の機械学習の手法に依存しない汎用的なアルゴリズムであるが，ここにニューラルネットワークによる強化学習・教師あり学習を適用したのが NFSP である．また，NFSP では教師あり学習のサンプリング方法を改善するために，通常の circular replay buffer ではなく reservoir buffer を用いている．これによって，有限のメモリでもデータ全体から等確率でサンプリングして保持することができ，平均戦略の計算を効率よく行えるようになっている．

3. 提案手法

本研究では，NFSP の教師あり学習が最適応答戦略の平均戦略を計算することを意図していることに着目し，教師あり学習の学習データに探索で得たデータを含めないようにすることを提案する．

NFSP の強化学習では，相手プレイヤの戦略に対する最適応答戦略を求めるために，探索と活用を行っている．具体的には，NFSP では ε-greedy を用いている．すなわち，確率 ε でランダムな手を選択し，確率 $1-\varepsilon$ で Q 値が最大であるような手を確定的に選択する．

ε-greedy を用いた Q 学習では，学習中は探索を含めた方策に基いて行動する．しかし，Q 学習は方策オフ型の学習方式で，Q 値が示す値は行動方策ではなく greedy に行動を選択した場合の期待累積報酬なので，最適応答戦略としては探索を含まない greedy な戦略を用いるべきである．探索を含めた方策を最適応答戦略として用い，平均戦略の計算に含めてしまうと，これがノイズとなって戦略がナッシュ均衡から外れてしまうことが考えられる．

オリジナルの NFSP では，ε-greedy の ε を学習の進行に合わせて小さくしているため，学習が進むにつれて探索を含めた戦略が greedy な戦略に近づいていき，先述の問題は起こらない．しかしながら，NFSP で行われる自己対戦は対戦相手の戦略が学習の進行に応じて変化するため，通常の Q 学習と異なり探索を無くすことはできないと考えられる．例えばある時刻 t について，状態 s で行動 a_1 を取る価値が a_2 よりも低いとすると，強化学習が正しく機能していれば平均戦略の a_1 を選択する確率は 0 に近づいていく．ここで，対戦相手の戦略が変化し，a_2 の価値は変化しないまま a_1 の価値が a_2 よりも高くなった場合，そのことを学習するためには s において a_1 を選択しなければならない．しかし，ε が十分小さくなってしまった場合，もはや a_1 が選択されることはなく，最適応答戦略を正しく求めることができなくなってしまう．

これらの問題を同時に解決するためには，探索の確率 ε をある程度大きな値に保ちつつ，活用によって得られたデータのみを教師あり学習に与えればよい．もし探索を含めた方策が最適応答戦略になっているのであれば，探索によって得られたデータを教師あり学習に含めないことで，学習が遅くなってしまう．しかし，ε-greedy を用いた Q 学習では，探索を含めた方策は一般に最適方策にはなっていないので，このようなことは起こらないと考えられる．

以上を踏まえた提案手法の擬似コードを Algorithm 1 に示す．提案手法に関わる部分は太字で示している．

4. 実験

提案手法の有効性を検証するために，小規模な不完全情報ゲームである 2 人 Leduc Hold'em に対し，提案手法を含むいくつか

Algorithm 1 改良 NFSP

```
Γ is a game and N is the number of players
function NFSP(Γ)
    Initialize Π, Q, θ^Π, θ^Q, θ^Q', M^SL, M^RL
    for i = 1, 2, ··· , N do                    ▷ 各プレイヤについて
        Initialize M_i
    end for
    for iteration = 1, 2, ··· do
        ε ← ε(iteration)                                  ▷ ε の値
        for i = 1, 2, ··· , N do
            σ_i ← ε-greedy(Q) (確率 η) or Π (確率 1 − η)
        end for
        Initialize Γ
        repeat
            n ← turn player of Γ
            M_n.s ← M_n.s'
            observe state s* and M_n.s' ← s*
            Store M_n in M^RL    ▷ 無条件でデータを保存
            if M_n.IsGreedy then   ▷ greedy に行動している強化学習由来のデータのみ保存
                Store M_n in M^SL
            end if
            Periodically update θ^Q with M ∼ M^RL , θ^Q'
            Periodically update θ^Π with M ∼ M^SL
            Periodically update θ^Q' ← θ^Q    ▷ RL のターゲットを更新
            Sample action a by strategy σ_i
            Execute a on Γ
            M_n.a ← a
            M_n.IsGreedy ← IsGreedy   ▷ 強化学習でかつ greedy に行動している場合のみチェック
            Reward M_n.r ← 0 ▷ 終端状態以外では報酬 0
        until Γ is over
        for i = 1, 2, ··· , N do
            set M_i.r ▷ 終端では状態に従って報酬を与える
        end for
    end for
    return Π (s, a|θ^Π)
end function
```

の設定で戦略を求め，得られた戦略の可搾取量 (exploitability) を計算し比較した．

Leduc Hold'em は以下のようなゲームである [Southey 05]．まず，3 種類のカードを各 2 枚ずつ，計 6 枚用意する．各プレイヤに 1 枚ずつ配り，場に 1 枚伏せて置き，残りの 3 枚は使用しない．各プレイヤは自分のカードだけを見ることができる．また，各プレイヤは場代として 1 単位のチップを供託する．ターンプレイヤは，賭けるチップを増やすベットか，相手と同じ賭け額にするコールか，ゲームから降りるフォルドを選択できる．ただし，一度もベットされていない場合はコール・フォルドは選択できず，代わりに何もしないチェックを選択できる．いずれを選択してもターンが相手に移り，お互いがチェックを選択するか，どちらかがコール・フォルドを選択するまでこれを繰り返す．この一連の流れをベットラウンドという．フォルドが選択された場合，フォルドを選択しなかったプレイヤが賭けられていたすべてのチップを得てゲームが終了する．それ以外の場合，伏せてあった場のカードを公開してもう一度ベットラウンドを行う．それでもフォルドが選択されなかった場合，お互いのカードを公開し，役が強い方のプレイヤが賭けられていたすべてのチップを得てゲームが終了する．ここで，役はワンペアまたはハイカードである．つまり，場のカードと同じ種類のカードを持っているプレイヤがいればそのプレイヤが勝利し，そうでない場合は数字の大きいプレイヤが勝利する．ベット時に増やすチップは，1 回目のベットラウンドでは 2，2 回目のベットラウンドでは 4 である．また，1 ベットラウンドにつきベットは 1 人 1 回しか選択できない．

可搾取量は，戦略プロファイルがナッシュ均衡戦略にどれだけ近いかを定量的に評価できる指標で，2 人零和不完全情報ゲームでは以下のように計算できる [Johanson 11]．

$$\epsilon(\sigma) = \sum_{p \in \{1,2\}} r_p(b_p(\sigma_{-p}), \sigma_{-p}) \tag{2}$$

すなわち，可搾取量はその戦略から最大で 1 ゲームあたりどれだけ搾取できるかを表している．この値は常に非負で，この値が 0 であることと戦略 σ がナッシュ均衡戦略であることは同値である．また，この値が 0 に近いほど，その戦略プロファイルはナッシュ均衡戦略に近いと言える．可搾取量を計算するためにはゲーム木を全探索する必要があるため，大規模なゲームに対しては可搾取量を計算することはできない．そのため，本論文では Leduc Hold'em という比較的単純化されたゲームを対象とした．

本研究では，教師あり学習の学習データに強化学習の探索で得たデータを含めないようにすることで，性能を悪化させることなく継続的に探索が行えるようになる，という仮説を立てている．そのため，比較する実験設定として以下の 4 つを用意した．

A. 通常の NFSP．ε は学習とともに急激に減衰する．具体的には，$\varepsilon = \frac{0.06}{\sqrt{n}}$，$n$ は [Heinrich 16] における iteration (128 game steps)．

B. A. において，教師あり学習の学習データに強化学習の探索で得たデータを含めないようにしたもの（提案手法）．

C. A. において，ε を減衰させず，固定したもの．

D. B. および C.，すなわち，教師あり学習の学習データに強化学習の探索で得たデータを含めないようにし，さらに ε を固定したもの．

ε の固定値には 0.1 を用いた．それ以外のハイパーパラメータは [Heinrich 16] に従っている．

それぞれの設定について，10^7 ゲーム行った後の可搾取量を 5 回計測した平均値を表 1 に示す．ε が減衰する設定では通常の NFSP と提案手法に差はないが，ε を固定値にした場合，通常の NFSP では性能が大きく悪化するが，提案手法では性能が有意に改善することがわかる．これは，通常の NFSP では探索の結果も平均に含まれるため，一定確率で探索を行うような設定では平均戦略にノイズが入ってしまうが，提案手法ではそのようなことが起こらず，学習が進んでも探索を続けられるからであると考えられる．

また，性能に対する ε の影響を調べるため，$0.01, 0.02, 0.05, 0.1, 0.2, 0.5$ の 6 種類の ε に対して同様の実験を行った．その結果を図 1 に示す．

既存の設定では，ε を固定すると性能が大きく悪化し，その度合いが ε の増加に従うことがわかる．これは，既存の設定

表 1: 各設定における，10^7 ゲーム行った後の可搾取量の値．各設定につき 5 回ずつ実験を行い，その平均を記載している．通常の NFSP (A.) と比較して，有意水準 $\alpha = 0.01$ で有意に差があるものには † を付けている．

設定	可搾取量
A.	$(3.8 \pm 0.5) \times 10^{-2}$
B.	$(4 \pm 1) \times 10^{-2}$
C.	$(26.8 \pm 0.8) \times 10^{-2\dagger}$
D.	$\mathbf{(2.3 \pm 0.5) \times 10^{-2\dagger}}$

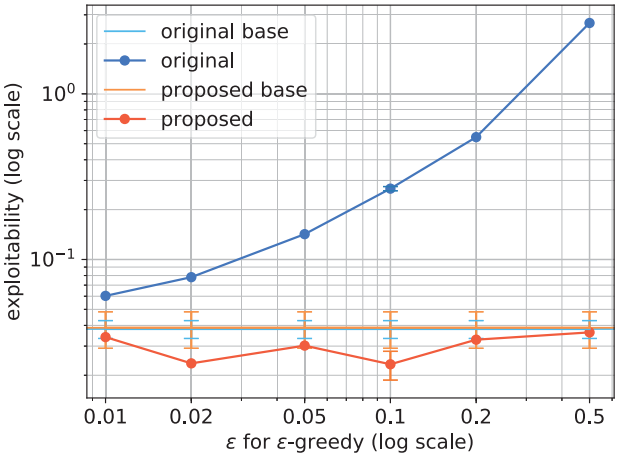

図 1: 各設定における，10^7 ゲーム行った後の可搾取量の値．名前に base が付く横線は，オリジナルの NFSP と同様，ε が急速に減衰する設定における結果を表している．$\varepsilon = 0.1$ の設定と ε が減衰する設定については 5 回ずつ実験を行った平均とその標準偏差を，それ以外のデータについては 1 回だけ実験を行った結果を表示している．

では探索の結果も平均戦略に含まれてしまうため，探索を行いすぎると性能が悪化する，という直感に即している．また，提案手法では，ε を固定すると性能が多少改善し，ε が 0.02 から 0.1 の間にある場合に特に性能に改善が見られるという大まかな傾向が見て取れる．これは，平均戦略の学習に探索を含めないため，探索を続けることで平均戦略の性能を落とすことなく最適応答戦略を正しく計算できるのではないかという直感に即している．ε が大きすぎる場合に性能が多少悪化するのは，活用で得られるデータが減少することで教師あり学習に学習させるデータの量が減ってしまい，教師あり学習の追従速度が低下するためであると考えられる．

5. おわりに

本研究では，NFSP の教師あり学習が最適応答戦略の平均戦略を計算することを意図していることに着目し，教師あり学習の学習データに強化学習の探索で得たデータを含めないようにすることで，性能を悪化させることなく継続的に探索が行えるのではないか，という仮説を立て，この手法によって NFSP の性能を向上させることができることを実験的に示した．

NFSP は，一般の不完全情報ゲームに対し強化学習を用いて事前知識のない状況から解くことができるという汎用的な手法である．今後も NFSP の性能を向上させる研究を続けるとともに，実際に NFSP を種々の大規模な不完全情報ゲームに対して適用することも行っていきたい．

参考文献

[Bowling 15] Bowling, M., Burch, N., Johanson, M., and Tammelin, O.: Heads-up limit hold'em poker is solved, *Science*, Vol. 347, No. 6218, pp. 145–149 (2015)

[Brown 51] Brown, G. W.: Iterative solution of games by fictitious play, *Activity analysis of production and allocation*, Vol. 13, No. 1, pp. 374–376 (1951)

[Brown 17] Brown, N. and Sandholm, T.: Superhuman AI for heads-up no-limit poker: Libratus beats top professionals, *Science* (2017)

[Heinrich 15] Heinrich, J., Lanctot, M., and Silver, D.: Fictitious Self-Play in Extensive-Form Games, in *Proceedings of ICML*, pp. 805–813, JMLR Workshop and Conference Proceedings (2015)

[Heinrich 16] Heinrich, J. and Silver, D.: Deep Reinforcement Learning from Self-Play in Imperfect-Information Games, *arXiv:1603.01121* (2016)

[Johanson 11] Johanson, M., Waugh, K., Bowling, M., and Zinkevich, M.: Accelerating Best Response Calculation in Large Extensive Games, in *Proceedings of the 22nd IJCAI - Volume 1*, pp. 258–265 (2011)

[Mnih 15] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D.: Human-level control through deep reinforcement learning, *Nature*, Vol. 518, pp. 529–533 (2015)

[Moravčík 17] Moravčík, M., Schmid, M., Burch, N., Lisý, V., Morrill, D., Bard, N., Davis, T., Waugh, K., Johanson, M., and Bowling, M.: DeepStack: Expert-level artificial intelligence in heads-up no-limit poker, *Science* (2017)

[Southey 05] Southey, F., Bowling, M., Larson, B., Piccione, C., Burch, N., Billings, D., and Rayner, C.: Bayes' Bluff: Opponent Modelling in Poker, in *Proceedings of the Twenty-First Conference on UAI*, UAI'05, pp. 550–558, Arlington, Virginia, United States (2005), AUAI Press

[Tammelin 14] Tammelin, O.: Solving Large Imperfect Information Games Using CFR+, *arXiv:1407.5042* (2014)

[河村 17] 河村 圭悟, 鈴木 潤, 鶴岡 慶雅：未知環境における多人数不完全情報ゲームの戦略計算, ゲームプログラミングワークショップ 2017 論文集, pp. 80–87 (2017)