2019年度　修士論文

対戦ログに基づいた
多様な戦略を持つポーカーAIの構築

電気通信大学大学院
情報理工学研究科
情報学専攻

令和2年1月27日

1830047　　小山 祐希
主任指導教員　　橋山 智訓 准教授
指導教員　　田野 俊一　　教授

目 次

第1章　はじめに

1.1　背景

AI技術の発展によりゲームAIは様々なゲームにおいて、人間プレイヤーを超える強さを獲得しつつある。プレイヤーがゲームの全ての状態を観測できる完全情報ゲームでは、チェスや将棋などのボードゲームで人間のプロ選手を破るAIが開発されている [1] [2]。探索空間が広く人間に勝つことは難しいとされてきた囲碁でも、2015年にAlphaGoがトップレベルのプロ選手を相手に勝利を収めた [3]。また、近年では不完全情報ゲームに関する研究も盛んに行われており、国内でもパーティーゲームとして人気のある「汝は人狼なりや？」や、トランプゲーム「大貧民」のAIの競技会が開催されている [4] [5]。

一方でコンピュータゲームの普及により、強いプレイヤー相手に勝つことだけを目的としないAIの研究も進んでいる。例えばプレイヤーのレベルに合わせて対戦を行うAIや、オンラインゲームにおいて不足しているプレイヤーの代理となるAIなど、人間プレイヤーを楽しませるためのAIの需要が高まりつつある。これらのAIは強さだけでなくプレイヤーを飽きさせないための性格付けや、違和感を覚えないような人間らしさが求められている。しかし従来のこれらのAIは、人の手によって強さの調整や性格付けが行われることが多く、設計に大きなコストが必要とされる。そこで近年では遺伝的アルゴリズムや強化学習を用いて、パラメータ調整を自動で行う研究が行われている [6] [7]。

1.2　目的

人を楽しませるためゲームAIにおけるパラメータ調整の自動化に関して、改善すべき重要な点として次の3つが考えられる。一つは調整対象となるパラメータを選定するのにかかるコストである。異なる2人の人間プレイヤーは同じ環境下でも異なる行動を選択し得る。しかしプレイヤー間の戦略の違いを定量的に表すことは困難である。同様に、異なる戦略を持つAIの違いを、パラメータという形で表すことも困難な作業である。また、良いパラメータを設定するには対象ゲームに対する深い理解が必要である。同じ手法を異なるゲームに実装し直す際にも再び人手に

よる作業が必要となる。2 つ目は調整対象であるパラメータから実際に AI を作成するための、アルゴリズムの最適化の困難さである。パラメータはベクトルであり、ゲームの環境を観測して行動を選択するには、パラメータを利用するアルゴリズムが必要である。例として決定木によって行動を決定するアルゴリズムにおける、各分岐点の条件式の閾値をパラメータとするような手法が考えられるが、木構造の最適化については確立された手法が存在しない。また、AI を性格付けによっていくつかのグループに分類する場合、特徴量への重み付けが必要であることがみにくいアヒルの子定理として知られているが、重み付けには人の知識が必要とされる。3 つ目はパラメータ調整によって生成された多様な AI の戦略を、人間プレイヤーから乖離させないことである。特にオンラインゲームなどで人間プレイヤーの代わりとなる AI の場合、一般的な人間プレイヤーからかけ離れた戦略を持つことは、プレイヤーに違和感を与える原因となる。ゲーム理論の分野では人は不確実性を含む選択肢に直面した際に、それぞれの結果が起こる確率とそのときの利得をかけ合わせた期待効用を、最大化するように合理的に行動すると仮定されている。しかし実際に人間が意思決定を行う際には確率や利得にそれぞれ認知バイアスが働く。期待効用が認知バイアスに影響されることで、人間は状況に応じてリスク回避的もしくはリスク指向的に振る舞う。これを考慮し、人間プレイヤーに近い戦略に制限してパラメータ調整を行うには、人間の認知やゲームへの深い理解、過去のプレイヤーのログの分析が必要となり、多大なコストを要する。

本研究ではこれらの課題を解決するために、ゲームにおけるプレイヤーのログから類似の戦略を分類するとともに、ニューラルネットワークを用いて各クラスタを代表する戦略を学習する。提案手法ではゲーム環境を直接入力して行動を出力できるため、人の知識によるパラメータの選定や行動アルゴリズムの設計が必要ない。実際に 2015 年に開発された Deep Q-Network [8] は、ゲーム画面とスコアのみを入力とし、行動を出力とするディープニューラルネットワークと、強化学習を組み合わせたエージェントにより、Atari 2600 に含まれる 49 種のゲームのうち 29 種で、プロの人間プレイヤーと同等以上のパフォーマンスを達成している。また、遺伝的アルゴリズムのように乱数によって多様性を獲得するのではなく、人間プレイヤーのログに基づく戦略の分類によって複数の AI を生成することにより、人間プレイヤーの戦略から乖離した戦略を持つ AI が生成されることを防ぐ。本研究では対象のゲームとして不完全情報ゲームの一種であり、世界的に広くプレイされているテキサスホールデムポーカーを用いて、提案手法の有効性の検証を行う。

1.3 本論文の構成

本論文は全 6 章で構成される。第 2 章では本研究の対象となるテキサスホールデムポーカーについて、ルールの説明と一般的な戦略の区分について述べる。第 3 章では関連研究として、ゲームにおけるプレイヤーの戦略の分類や、多様性をもつゲーム AI や相手によって戦略を変化させる AI に関する研究を紹介し、その問題点についても述べる。また、人間の認知バイアスに関する研究として、行動経済学における人間の意思決定モデルである、プロスペクト理論についても言及する。第 4 章では戦略の分類と学習について具体的な手法を説明し、第 5 章で有効性の検証のため行った実験の結果を述べる。第 6 章では実験の結果をもとにまとめと今後の課題について述べる。

第2章　テキサスホールデムポーカー

本研究では対象のゲームとしてテキサスホールデムポーカーを取り扱う。テキサスホールデムはカジノゲームであるポーカーのルールの一つであり、世界的に広くプレイされている。また、テキサスホールデムは不完全情報ゲームであり、近年ではゲーム研究の分野で度々研究の対象となっている。2017 年には Norm ら [9] によって作成された「Libratus」が、ベット額に上限がない一対一のテキサスホールデムにおいて、4 人のトッププロ選手に勝利した。本章ではテキサスホールデムの基本的なルールと、一般的な戦略の区分について述べる。

2.1　基本的なルール

2.1.1　ゲームの流れ

テキサスホールデムのテーブルでは図 2.1 のように、カードやチップが配置される。まず各プレイヤーにはホールカードと呼ばれる、他のプレイヤーには非公開の 2 枚の手札が配られる。また一人のプレイヤーの前にはディーラーを示すボタンが置かれる。このボタンは勝者が決定し新たなゲームが開始される際に時計回りに移動する。ここから 1 回目のラウンドが開始され、まずボタンの左隣のプレイヤーが、スモールブラインドと呼ばれる強制ベットを支払い、更にその左隣のプレイヤーが同様にビッグブラインドと呼ばれる強制ベットを支払う。スモールブラインドはビッグブラインドの半額であり、額はゲームごとに決定されている。ビッグブラインドの左隣のプレイヤーから、時計回りで各プレイヤーのターンが回っていき、プレイヤーは自身のターンで後述する行動のいずれかを選択する。ラウンドはターンが一周した後に全員のベット額が同額になった時点で終了する。ラウンドが終了すると、ベットされたチップをポットに回収してから次のラウンドに移行し、最大 4 回目のラウンドまで継続する。2 回目以降のラウンドでは強制ベットはなく、ボタンの左隣のプレイヤーから時計回りでターンが進行する。また、ラウンドが進む毎にテーブル中央に、コミュニティカードと呼ばれる全プレイヤー共有のカードが 3 枚、4 枚、5 枚とオープンされる。途中でゲームの参加者が一人になった場合はその時点で残ったプレイヤーが勝利となり、4 回目のラウンドが終了した時点で 2 人以上の

図 2.1: テキサスホールデムをプレイ中のテーブルの様子 [10]

プレイヤーが残っていた場合は、各プレイヤーのホールカードを公開し、後述する役のうち最も強い役を作ったプレイヤーの勝利となる。勝利したプレイヤーはポットを全て獲得する。

2.1.2 賭けの行動

各ラウンドでプレイヤーが自分のターンに取ることができる行動を以下に挙げる。

- チェック: チップを賭けずにパスしてゲームに残る。そのラウンド内でまだチップが賭けられていないときのみ宣言できる。
- コール: 現在のベット額と同額になるようにチップを賭けてゲームに残る。
- ベット、レイズ: ベット額を上乗せしてゲームに残る。チップが賭けられていない状態のときにベット、すでに賭けられている額を吊り上げるときにレイ

ズを宣言する。上乗せする額はルールによって異なるが、本研究ではリミットというルールを採用する。このルールでは 1 回目と 2 回目のラウンドではビッグブラインドと同額、3 回目と 4 回目ではビッグブラインドの倍の額を、ベットまたはレイズするごとに上乗せする。さらに 1 回のラウンドにつきベットとレイズは計 4 回までの制限がつく。

- フォールド: チップを賭けずにゲームから降りる。ゲームから降りたプレイヤーは以降のラウンドに参加できず、それまでに賭けたチップも戻ってこない。フォールドはそのラウンド内でまだチップが賭けられていないときでも、ルール上は宣言できるが、チェックに対して優位な点がないので本研究では不可能とする。

2.1.3 役の強さ

テキサスホールデムでは各プレイヤーに配られる 2 枚のホールカードの他に、テーブル中央に配置されるコミュニティカードを使って役を決定する。4 回目のラウンドが終了した時点で 2 人以上のプレイヤーがゲームに残っていた場合、2 枚のホールカードと 5 枚のコミュニティカードの、計 7 枚のうち任意の 5 枚から作れる役の強さで勝者を決定する。以下にテキサスホールデムの役を強い順に挙げる。

1. ストレートフラッシュ: 5 枚の同じスートで連続する数字のカードからなる。K と A は基本的に連続しないが 10-J-Q-K-A では役が成立する。ストレートフラッシュ同士の勝負では、最もランクの高いカードのランクが高い方が強い。ランクとはカードの数字に対応する強さで、全ての役において A が最も高ランクであり、他は数字が大きいほど高ランクである。

2. フォーカード: 4 枚の同じ数字のカードと 1 枚の任意のカードからなる。フォーカード同士の勝負では 4 枚のランクが高い方が強く、同一の場合は残った 1 枚のランクが高い方が強い。

3. フルハウス: 3 枚の同じ数字のカードと、2 枚の同じ数字のカードの組み合わせからなる。フルハウス同士の勝負では 3 枚のランクが高い方が強く、同一の場合は 2 枚のランクが高い方が強い。

4. フラッシュ: 5 枚の同じスートのカードからなる。フラッシュ同士の勝負ではランクの高いカードから順に比較する。

5. ストレート: 5 枚の連続する数字のカードからなる。ストレートフラッシュと同様に 10-J-Q-K-A では役が成立する。ストレート同士での強さもストレートフラッシュと同様である。

6. スリーカード: 3 枚の同じ数字のカードと 2 枚の任意のカードからなる。スリーカード同士の勝負では揃っている 3 枚のカード、残りの 2 枚のうちランクの高い方、残りのカードの順にランクを比較する。

7. ツーペア: 2 枚の同じ数字のカードの組み合わせ 2 セットと任意の 1 枚からなる。ツーペア同士の勝負では揃っている 2 枚の組み合わせのうちランクの高い方、ランクの低い方、残りのカードの順にランクを比較する。

8. ワンペア: 2 枚の同じ数字のカードと 3 枚の任意のカードからなる。ワンペア同士の勝負では揃っている 2 枚のカードを比較し、同一の場合は残りのカードをランクの高い順に比較する。

9. ハイカード: 上述の役がいずれも満たされなかったときの役である。ハイカード同士の勝負ではランクの高いカードから順に比較する。

いずれの役でもスート間の強弱関係はなく、全く同じ数字で構成される役同士の勝負は引き分けとなり、獲得するチップは山分けとなる。

2.2 テキサスホールデムにおける戦略

テキサスホールデムでは一般的にプレイヤーの戦略は、図 2.2 の様にルース/タイト、アグレッシブ/パッシブの 2 つの指標によって分類される。これらを組み合わせた 4 つのタイプが存在することが知られている [11]。ルースはフォールドをあまり選択せず、多くのゲームをプレイするプレイヤーを指し、タイトは反対に多くのゲームで早めにフォールドするプレイヤーを指す。アグレッシブは参加したゲームにおいて、ベットやレイズを多用してベット額を吊り上げるプレイヤーを指す。対してパッシブはチェックやコールでゲームに残るプレイヤーを指す。一般的にパッシブなプレーはあまり好まれずアグレッシブの方が強いとされている。

このようにテキサスホールデムでは戦略のスタイルが複数あることが認知されている。戦略の分類を行う際の指標としてわかりやすいことも、本研究の対象のゲームとしてテキサスホールデムポーカーを選択した理由の一つである。

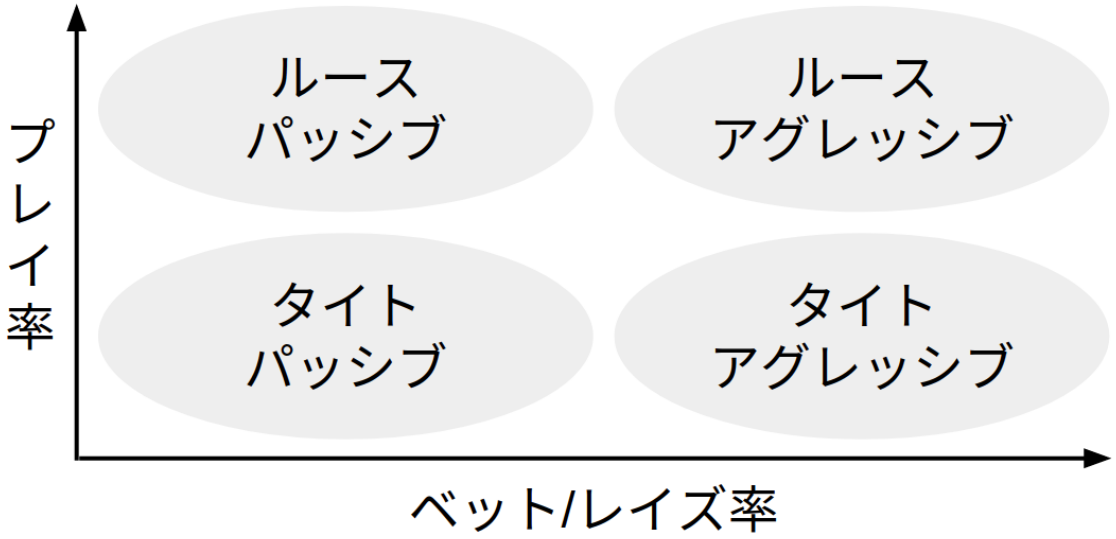

図 2.2: テキサスホールデムにおける一般的な戦略の分類

第3章　関連研究

3.1　戦略の分類と予測

主にコンピュータゲームのゲーム設計において、プレイヤーが実際にどのように行動するかを予測することは重要な過程である。そこでプレイヤーの行動ログから選択されたいくつかの特徴量を利用して、プレイヤーの分類や行動の予測を行う研究が行われている。

Bauckhage ら [12] はゲームプレイヤーから得られる大量の行動データを、適切にゲームの改善に役立てるために、ゲーム研究におけるクラスタリングについて、データの特徴や目的ごとの最適な手法についての検討を行った。この研究では、データの次元数や目的、データ同士の関係性などの観点から、最適なクラスタリング手法の提案を行った。さらに実際にアクションゲーム内の物理的な空間のクラスタリングを、複数の手法で行った結果を示している。Drachen ら [13] は、ゲーム開発におけるテストプレイの一部自動化を目的に、プレイヤーを行動データに基づいて分類する実験を、商用ゲームである「Tomb Raider: Underworld」で行った。この実験では自己組織化マップを使用して、プレイヤーをゲーム完了時間や死亡数、死因、ヒント要求回数などの特徴量を元に分類した。実験の結果、プレイヤーは 4 つのクラスタに分類され、この情報はゲーム内でどの程度のプレイヤーが、開発者の意図通りの行動を取っているかを評価するために活用できるとしている。

また、Wang ら [14] は、ゲーム環境の設計の変更がプレイヤーの行動にどのようの影響するかを調べるため、プレイヤーの価値構造をモデル化する手法を提案した。この手法ではプレイヤーがゲームをプレイする動機は勝利することだけではなく、多面的な動機づけがあるという過程の下で、逆強化学習を用いて各動機の重み付けを最適化している。実際にオンラインゲーム「World of Warcraft」における、ゲームプレイの動機を以下のように定義した。

- 進歩: レベルアップの速度
- 競争: 人間プレイヤーと競うための施設を訪れた数
- 人間関係: 現在のギルドに所属している期間

- チームワーク: チームの機能が備えられた施設を訪れた数
- 現実逃避: ゲームへの接続時間と連続ログイン日数の線型結合

これら5つを仮定し実験を行ったところ、上位プレイヤーと下位プレイヤー、ゲーム内でグループに所属しているプレイヤーとしていないプレイヤーなどの間で、価値構造が有意に異なることが明らかとなり、ゲーム環境が変更された際のプレイヤーの価値構造の変化も示された。

これらの手法では、分類や予測のためにプレイヤーの行動データから、適切な特徴量を選定して使用する必要がある。対象のゲームに対する深い知識が必要とされ、他のゲームに適用する際にも、特徴量を選定し直す手間がかかる点が問題として挙げられる。

3.2 戦略の多様性

プレイヤーを楽しませることを目的に、戦略に多様性を持たせて複数のAIを生成した研究について取り上げる。Esparcia-Alcázarら[6]はシューティングゲームにおいて、多様なタイプの振る舞いをするボットの獲得を行った。この研究では遺伝的アルゴリズムの適応度関数を変化させることで、最終的に得られるボットがどのように変化するかを検証する実験を行った。適応度関数を敵を倒した数、生存時間、倒した数と生存時間バランス型の3種類で実験を行ったところ、生存時間を適応度関数とした実験では、ゲームに搭載された標準的なボットよりも優れた成績を収めた。しかしながら、より優れたボットを作るにはさらに複雑な適応度関数が必要で、それは必ずしも目的を直接含むものではないと結論づけている。

上田ら[15]はオセロにおいてプレイヤーと同程度の実力を持つ、多様なAI郡を構成する手法を提案した。この手法では遺伝的アルゴリズムによってAIの実力をプレイヤーに近づけるが、同時にAI同士の類似度を適応度から減算することによって、最終的なAI郡が類似する戦略を獲得しないようにしている。この手法により得られたAI郡の中から、複数のペアを作って被験者に区別させる実験を行ったところ、中級者プレイヤーからは高い精度で同じAIのペアと異なるAIのペアが判別された。

福嶋ら[7]は2Dアクションゲームにおいて、先験的な情報を与えずに個性が表出するNPCを獲得する手法を提案した。この手法ではノンプレイヤーキャラクター（NPC）の行動を制御する決定木中の、各条件式の閾値を島モデルGAと呼ばれる遺伝的アルゴリズムによって最適化することにより、目標とする強さに合った個体を生成する。島モデルGAは、複数のほとんど独立した遺伝的アルゴリズムの計算を並列に行う手法であり、異なる振る舞いをする複数の個体を獲得することができ

る。実験の結果得られた NPC の振る舞いの差異は、主観評価実験によって「慎重型」や「積極型」のような個性として解釈された。

この様に、戦略の多様性を持たせることを目的とする研究では、遺伝的アルゴリズムが多く利用されている。しかしこれらはルールベースで行動する AI において、ルール内の条件分岐の閾値を遺伝的アルゴリズムによって最適化する手法を取っており、AI の行動ルール生成の自動化や最適化には至っていない。

3.3 戦略の適応

対戦相手の戦略に応じて AI 側の戦略を変化させる研究について取り上げる。杉本ら [16] は対戦型パズルゲームにおいて、AI プレイヤーの実力を人間プレイヤーと互角となるように調節する手法を提案した。この手法では対戦相手の行動が、事前に用意した複数の AI とどの程度戦略が一致しているかを推定し、推定結果をもとに戦局が互角になるような行動を取る。実験の結果、作成された AI プレイヤーはランダムに行動する AI と比較して、平均試合時間は長くなったが勝率はあまり改善されなかったとしている。

小野ら [17] はエアホッケーシミュレータを用いた研究において、対戦相手の戦略に応じて勝つための戦略を構築し、適宜切り替えながら対戦を行うモデルを提案した。この手法では、事前に異なる特徴を持つ 4 種の戦略を持つ対戦相手に対して、それぞれ Q 学習の派生である Q-PSP 学習を行って Q 関数を生成し、対戦時に相手に応じて 4 つの Q 関数の中から適切なものを選択して使用している。提案モデルは対戦実験において、学習に使用したものと同じ 4 種の戦略を切り替えながら戦う相手に対して、相手が戦略を切り替える毎に自身の戦略をスムーズに切り替え、失点を抑えつつゲームに勝利した。

これらの手法では事前に用意した AI に依存して自身の行動を決定しているため、別のゲームに適応させるためには人の手で AI を新たに作成しなければならない。また様々な戦略の相手に適応できるように AI を用意するには、対象のゲームに対する知識や、人間プレイヤーの過去の行動パターンの分析などが必要となる。

3.4 プロスペクト理論

3.1 や 3.3 では、プレイヤー毎の行動予測や対戦相手への適応に関する手法を紹介したが、人の意思決定全般に関する研究が心理学や行動経済学の分野で行われている。Kahneman ら [18] [19] はリスク下や不確実性の下での意思決定モデルとして、プロスペクト理論およびその拡張である累積プロスペクト理論を提唱した。これら

の理論では人は富の状態ではなくその変化から効用を得るが、そこに 2 つの認知バイアスがかかるとしている。

一つは実際の価値に対する主観的な価値で、価値関数とよばれる関数

$$v(x) = \begin{cases} x^{\alpha} & (x \geq 0) \\ -\lambda(-x)^{\beta} & (x < 0) \end{cases} \tag{3.1}$$

で表される。25 名の大学院生を対象に行われた選択実験の結果、パラメータ α, β の値は回帰分析により共に 0.88、λ は 2.25 と推定された。この関数の概形は図 3.1 に示すとおりであり、これは利得と損失の判断の基準点が、0 や正ときはリスク回避的に、負のときはリスク指向的になることを意味する。例として「1/2 の確率で 10 の利得が得られ 1/2 の確率で 10 の損失を被る」ギャンブルがある場合、多くの人は $10^{\alpha}/2 + (-\lambda) \times 10^{\beta}/2 < 0$ であるためこのギャンブルには参加しない。また「1/2 の確率で 20 の利得が得られ 1/2 の確率で何も得られない」か、「必ず 10 の利得が得られる」を選択できる場合、$20^{\alpha}/2 < 10^{\alpha}$ であるため多くの人はギャンブルを避け後者を選ぶ。一方、「1/2 の確率で何も得ず 1/2 の確率で 20 の損失を被る」か、「必ず 10 の損失を被る」を選択できる場合、$-\lambda \times 20^{\beta}/2 > -\lambda \times 10^{\beta}$ となり、多くの人は前者のギャンブルを選択する。

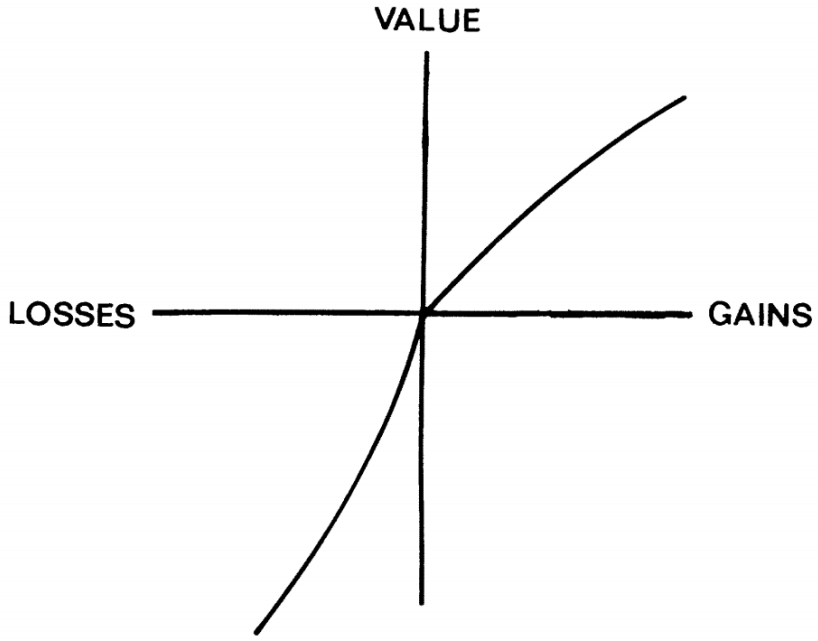

図 3.1: 価値関数 [18]

もう一つは実際の確率に対する見込みで次の関数

$$w^+(p) = \frac{p^\gamma}{(p^\gamma + (1-p)^\gamma)^{1/\gamma}} \tag{3.2}$$

$$w^-(p) = \frac{p^\delta}{(p^\delta + (1-p)^\delta)^{1/\delta}} \tag{3.3}$$

で表される。w^+, w^- はそれぞれ得られる利得が正の場合と負の場合に対応する関数であり、γ, δ はそれぞれ $0.61, 0.69$ と推定された。この関数の概形は図 3.2 に示すとおりであり、0 に近い確率を過大評価し 1 に近い確率を過小評価することを示している。

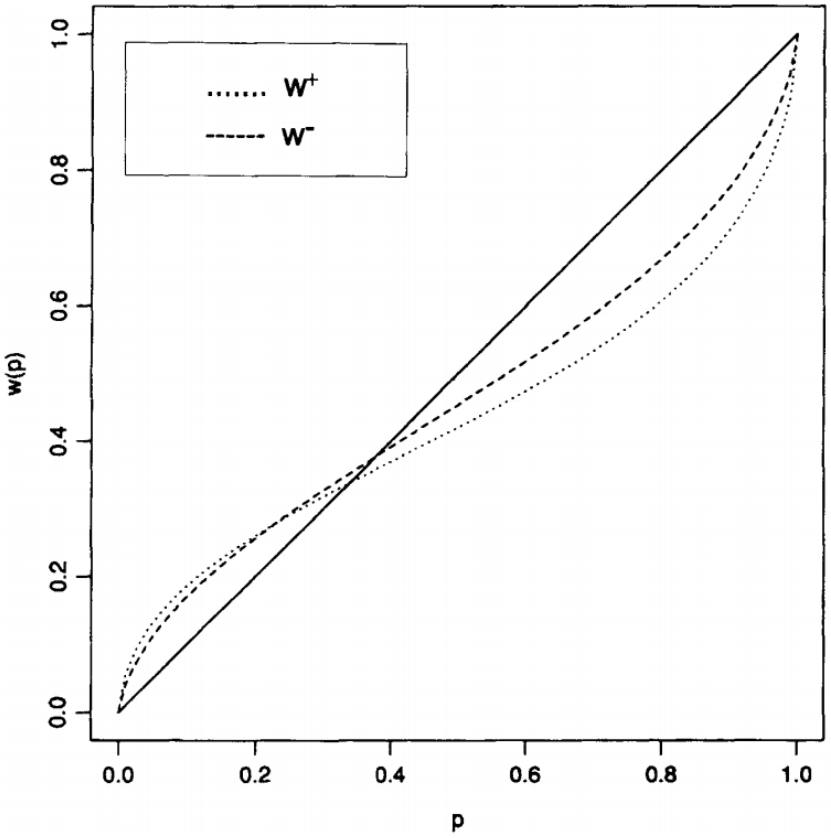

図 3.2: 決定荷重関数 [19]

プロスペクト理論は行動経済学の分野で提唱された理論であり、ゲーム研究において直接適用した例は少ないが、人の意思決定モデルを適用することで、AI の振る舞いを実際の人間プレイヤーに近づけられる可能性がある。

第4章　提案手法

本研究ではゲーム固有の知識を使わずに、対戦ログに基づいた多様な戦略を持つAIを生成することを目標とする。提案手法は戦略の分類と学習という2つの機能を持つ。本章ではこれらのアルゴリズムについて説明する。

4.1　戦略の学習

ニューラルネットワークはヒトの脳の神経の機能をもとに作られた数学モデルであり、今日ではゲーム研究や画像処理、言語処理など多岐にわたる分野で活用されている。テキサスホールデムではニューラルネットワークを活用したDeepStackが、プロプレイヤーを破る成果を挙げている[20]。本研究では戦略を、ゲーム環境を入力として行動を出力とする関数と捉え、これを図4.1のように同様の入出力を持ち、信号が入力層から中間層、出力層へ伝播する、階層型の順伝播型ニューラルネットワークによって学習する。このニューラルネットワークを使用し、ゲーム環境と選択された行動のログを訓練データとして学習させることで、テキサスホールデム固有の知識を使わずにAIを生成することが可能である。

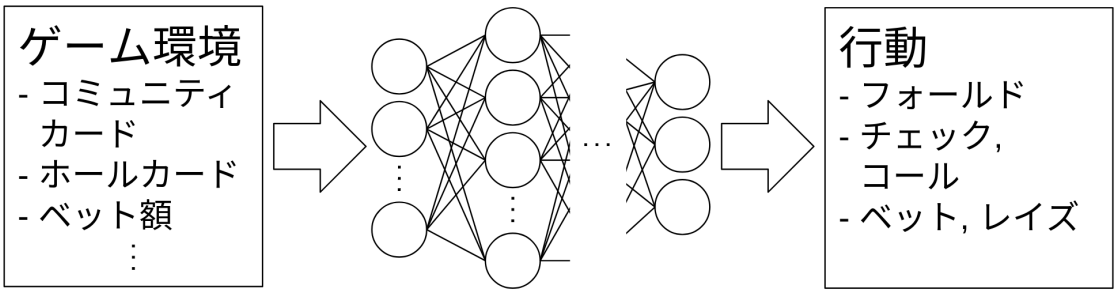

図4.1: 順伝播型ニューラルネットワーク

4.2 戦略の分類

4.2.1 k-means 法と自己組織化マップ

k-means 法は非階層型クラスタリングのアルゴリズムの一つであり、現在では最も一般的なクラスタリング手法の一つである [21]。*k*-means 法の手順の例を以下に示す。

1. n 個のデータ $x_i(i = 1, ..., n)$ に対応するラベル $l_i(i = 1, ..., n)$ を $[1, k]$ の範囲の整数でランダムに初期化する。

2. クラスタ中心 $v_j(j = 1, ..., k)$ を $l_i = j$ を満たす全ての x_i の中心へと更新する。中心の計算は一般的に算術平均で行われる。

3. 各 x_i に対して全ての v_j との距離を計算し、最も近いクラスタ中心のインデックスを l_i に記録する。

4. クラスタの変化が収束するまで 2 と 3 を繰り返す。

k-means 法は目的関数式 (4.1)

$$\sum_{i=1}^{n} \min_{j \in \{1,...,k\}} d(x_i, v_j) \tag{4.1}$$

を最小化する、NP 困難な問題の局所解を求めるヒューリスティックなアルゴリズムである。*k*-means 法はゲーム研究の分野も含め広く使われている一方で、図 4.2 のようにガウス分布に従わないデータでは、適切な結果が得られない場合がある。またゲームのプレイ履歴のような高次元データでは、次元の呪いが発生することにより良い結果を得ることが難しい。

また *k*-means 法と類似したモデルとして、自己組織化マップ（Self-organizing maps, SOM）が挙げられる。SOM は以下の手順で入力の低次元空間への写像を得る。

1. k 個の入力と同じ次元の重みベクトル $v_j(j = 1, ..., k)$ をランダムに初期化する。各 v_j はその値とは別に座標を持ち、二次元か三次元空間程度の空間に格子状やハニカム構造などで並べられる。

2. n 個の入力ベクトル $x_i(i = 1, ..., n)$ 全てに対して、最も距離の小さい v_j を計算する。

3. 各 x_i に対して、2 で求めた重みベクトルを学習率に従って x_i に近づける操作をする。このとき 2 で求めた重みベクトルと座標が近い重みベクトルも、同時に少し x_i に近づける。学習率や影響の及ぶ重みベクトルの範囲は繰り返しとともに減少する。

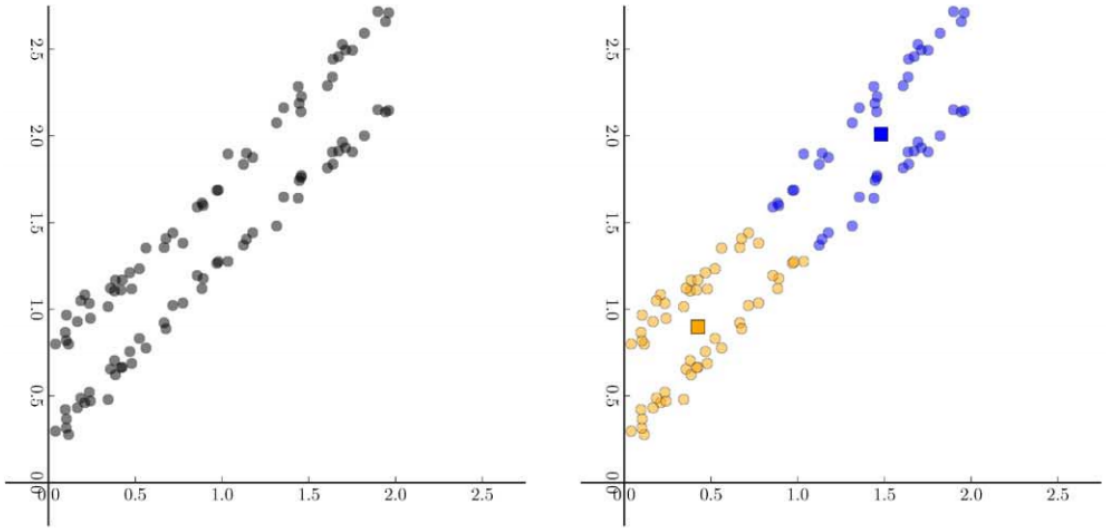

図 4.2: k-menas 法の失敗例 [12]

4. 2 と 3 を規定回数繰り返す。

重みベクトルを更新する操作は暴露とも呼ばれ、暴露が最近傍の重みベクトルだけでなく周囲の重みベクトルにも影響することにより、類似する重みベクトルが低次元空間上で隣接する形に収束する。SOM は各 x_i に対して最も距離の近い v_j を求めて各 v_j を更新するという点で、*k*-means 法と類似している。実際に手順 3 の重みベクトルの更新をまとめて行うバッチ学習 SOM では、暴露の影響する範囲を 0、学習率を 1 に設定した場合、*k*-means 法と等しい更新が行われる。

4.2.2 学習器を用いた分類

ゲームプレイヤーの戦略の分類は、行動ログから特徴量を抽出し、*k*-means 法などのクラスタリング手法を適用することで行われてきた。しかし適切な特徴量の抽出にはゲーム固有の知識が必要で困難な上に、対象となるゲーム毎に人の手で再構成する必要がある。さらに特徴量が高次元であれば、次元の呪いが発生しクラスタリングの精度が落ちる可能性があるため、必要な特徴量を残しつつ次元削減を行う必要がある。

提案手法では特徴量の抽出を行わずに、ニューラルネットワークなどの学習器を使用して、対戦ログ中のゲーム環境と行動のペアから直接戦略の分類を行う。この手法では図 4.3 に示した概要図のように、プレイヤー毎に分けられた対戦ログと複数の学習器によって戦略の分類を行う。具体的な分類の手順を以下に示す。

表 4.1: k-means 法と SOM、提案手法の比較

	k-means 法	SOM	提案手法
入力データ	ベクトル	ベクトル	環境と行動のペアの集合
v_j	ベクトル	ベクトル	学習器
ラベルの更新	最小距離	最小距離	最大予測精度
v_j の更新	算術平均の計算	暴露	学習器の学習

1. k 個の学習器 $v_j(j = 1, ..., k)$ の重みを初期化し、n 人のプレイヤーの対戦ログ $x_i(i = 1, ..., n)$ に対応するラベル $l_i(i = 1, ..., n)$ を $[1, k]$ の範囲の整数でランダムに初期化する。

2. 各学習器 v_j に対して、$l_i = j$ を満たす全ての x_i を訓練データとして学習を行う。

3. 各 x_i に対して全ての v_j で予測を行い、最も予測精度の高い学習器のインデックスを l_i に記録する。

4. 各学習器の学習が収束するまで 2 と 3 を繰り返す。

各学習器は予測精度の低い対戦ログでは学習を行わず、訓練データを予測精度の高いものに絞って適応していく。これにより類似した戦略を持つプレイヤーの対戦ログが、同一の学習器の訓練データとして使用されて、戦略が自動的に分類されることが期待される。また分類終了時の各学習器の持つ戦略が、そのままクラスタを代表する戦略となっているため、分類結果をもとに AI を構築するために行動ルールを再構築する必要がない。

提案手法の手順は k-means 法や SOM の手順に基づいており、その比較を表 4.1 に示す。データ構造の違いとして k-means 法や SOM ではベクトルを入力とし、クラスタ中心や重みベクトルに入力と同次元のベクトルを設定する。しかし本研究での分類対象である戦略は、ゲーム環境を入力として行動を出力する関数であるので、同様の入出力を持つ学習器を使用して分類を行う。入力データとして関数を扱うことはできないので、代わりに戦略に対するいくつかの入力とその出力である対戦ログを使用する。ラベルの更新には距離に代わって予測精度を用いる。予測精度は訓練データの戦略と学習器の戦略が等価であるときに最大値の 1 を取るため、類似度の指標として有効であると考えられる。学習器の更新には対象の対戦ログを訓練データとして学習を行う。これは学習器を少しずつ訓練データに近づけていくという点で、SOM の暴露と類似の性質を持つ。

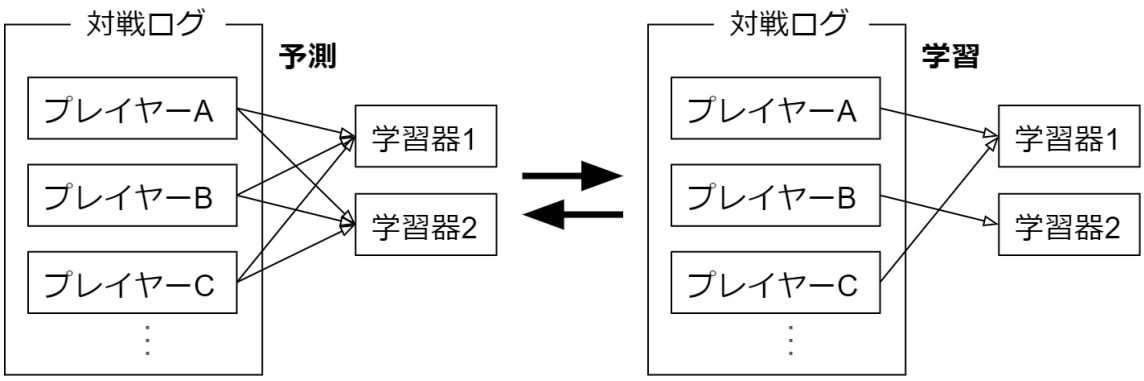

図 4.3: 分類システムの概要図

第5章　実験と結果

5.1　単純な関数の分類

提案手法の有効性を検証するため、ゲームへの適用の前段階として、いくつかのパラメータを持つ関数を提案手法によって分類し、その結果を確認する。基本となる関数を

$$f(x_1, x_2; p_1, p_2) = x_1^{p_1} + p_2 x_2 \tag{5.1}$$

とし、p_1, p_2 にそれぞれ $[0, 2]$ の範囲で 0.2 刻みの値を与え、計 121 個の関数を作成した。作成した 121 個の関数それぞれに、$[0, 5)$ の範囲の乱数で生成された x_1, x_2 を入力する操作を 128 回繰り返し、121×128 個の入力と出力のペアを生成した。これは 4.2.2 節で述べた分類手順における、プレイヤーの戦略を 121 個の関数に、各プレイヤーの対戦ログである環境と行動のペアの集合を、それぞれ 128 個の入力と出力のペアの集合に置き換えている。

学習器として表 5.1 に示す構造の順伝播型ニューラルネットワークを 4 つ作成し、生成したデータの分類を行った。この学習器は分類問題ではなく回帰問題を解くため、データのクラスタへの割り振りには予測精度ではなく誤差を利用している。各関数から生成されたデータが、最終的にどの学習器の訓練データとして割り振られたかを図 5.1 に示す。p_1 の値が大きい領域ではクラスタ 1 とクラスタ 3 が、ほとんど p_1 の値に依存して分類されており、p_1 の値が小さい領域ではクラスタ 2 とクラスタ 4 が、比較的 p_2 の影響を大きく受けて分類されている。これは式 5.1 の p_1 に関する偏微分 $x_1^{p_1} \log x_1$ が、p_1 が大きいほど大きくなる一方で、p_2 に関する偏微分 x_2 は p_2 の大きさにかかわらず一定であり、p_1 が大きい領域では p_1 の差異による影響が比較的大きいことから、妥当な結果と言える。これより提案手法は単純な関数の分類問題において、パラメータに沿って適切に関数を分類できることが明らかとなった。

表 5.1: 順伝播型ニューラルネットワークの構造

レイヤー数	4
ノード数	2-4-4-1
活性化関数	ランプ関数（ReLU）
最適化アルゴリズム	RMSProp
損失関数	平均二乗誤差

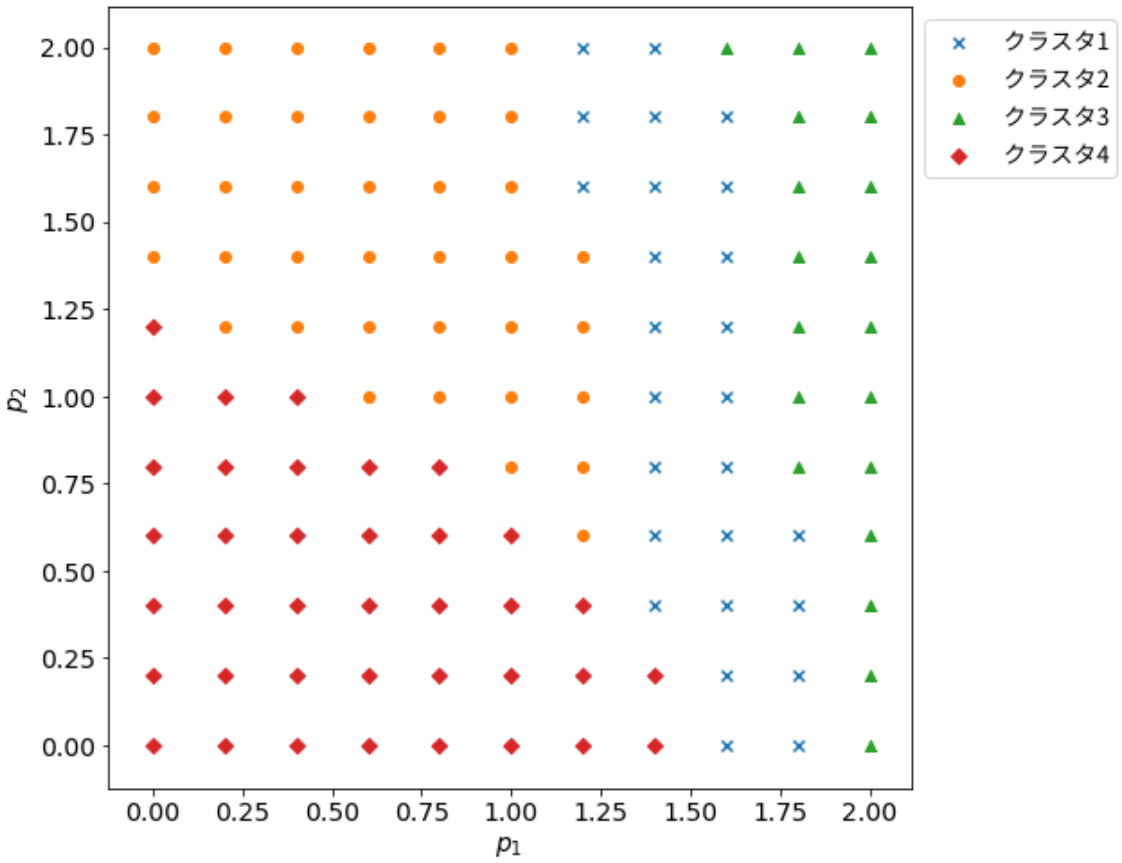

図 5.1: 各関数の分類結果

5.2 テキサスホールデムエージェントによる実験

5.2.1 エージェントの作成

テキサスホールデムにおける提案手法の有効性を検証するため、ルールベースの戦略を持つエージェントを複数作成し、エージェント同士の対戦ログから生成した訓練データにより学習を行った。エージェントはポーカー初心者の大学生と大学院生 10 名が行動ルールを設計し、計 12 個が作成された。これらのエージェントから毎回ランダムに 8 個を選択して、テキサスホールデムを行うシミュレーションを実行し、環境と行動のペアによる対戦ログを生成した。

対戦ログを生成する過程における各エージェントのスコアと、プレイ傾向を表 5.2 に示す。スコアはシミュレーション終了時の所持チップが、初期値から最も差の大きかったエージェントと A の絶対値を 1 として、各エージェントの最終的な所持チップの初期値との差をを正規化した値である。Fold、Check+Call、Bet+Raise は対応する行動を選択した回数である。チェックとコール、ベットとレイズはそれぞれ、常にどちらか片方しか選択できず、ベット額を上乗せするかどうかという点において等しい行動の組み合わせであるので、まとめて表記している。ルース度、アグレッシブ度はそれぞれ次の式によって求める。

$$\text{ルース度} = 1 - \text{Fold}/(\text{Fold} + \text{Check} + \text{Call} + \text{Bet} + \text{Raise}) \tag{5.2}$$

$$\text{アグレッシブ度} = (\text{Bet} + \text{Raise})/(\text{Check} + \text{Call} + \text{Bet} + \text{Raise}) \tag{5.3}$$

さらにルース度とアグレッシブ度を軸にとって、各エージェントのプレイスタイルをプロットしたところ図 5.2 のようになった。エージェント A と B は、全てのエージェントの中で 1 番目と 2 番目にスコアの高かったエージェントであり、極端にルースアグレッシブ、タイトアグレッシブなプレイスタイルを持つ。これらは最も単純なアルゴリズムを持つエージェントでもあり、A は全ての局面で可能な限りベットまたはレイズを選択し、B は 2 枚のホールカードの数字が等しければ可能な限りベットまたはレイズ、そうでなければフォールドを選択する。それ以外のエージェントのプレイスタイルはパッシブに偏っており、スコアが高いエージェントほどタイトな傾向にあった。エージェント C と E、G は必ずベットやレイズ以外の行動を選択するアルゴリズムを持つため、Bet+Raise の値が 0 になっている。

5.2.2 戦略の分類と行動予測

5.2.1 節で生成したログを用いて戦略の分類を行うため、ログに含まれるゲーム環境を表 5.3 に示す計 107 次元の特徴量に変換し、表 5.4 に示す構造の順伝播型ニュー

表 5.2: 各エージェントの対戦結果

エージェント	スコア	Fold	Check+Call	Bet+Raise	ルース度	アグレッシブ度
A	1	0	887	31178	100.00%	97.23%
B	0.692	4696	392	4350	50.24%	91.73%
C	0.526	4966	9208	0	64.96%	0.00%
D	0.432	4710	10635	1508	72.05%	12.42%
E	0.341	4502	11048	0	71.05%	0.00%
F	0.253	4586	6419	2041	64.85%	24.13%
G	0.233	3997	12411	0	75.64%	0.00%
H	-0.454	3682	19612	4337	86.67%	18.11%
I	-0.475	2819	20206	2012	88.74%	9.06%
J	-0.659	3782	14539	8125	85.70%	35.85%
K	-0.914	2636	28482	3213	92.32%	10.14%
L	-0.975	2862	21944	6041	90.72%	21.59%

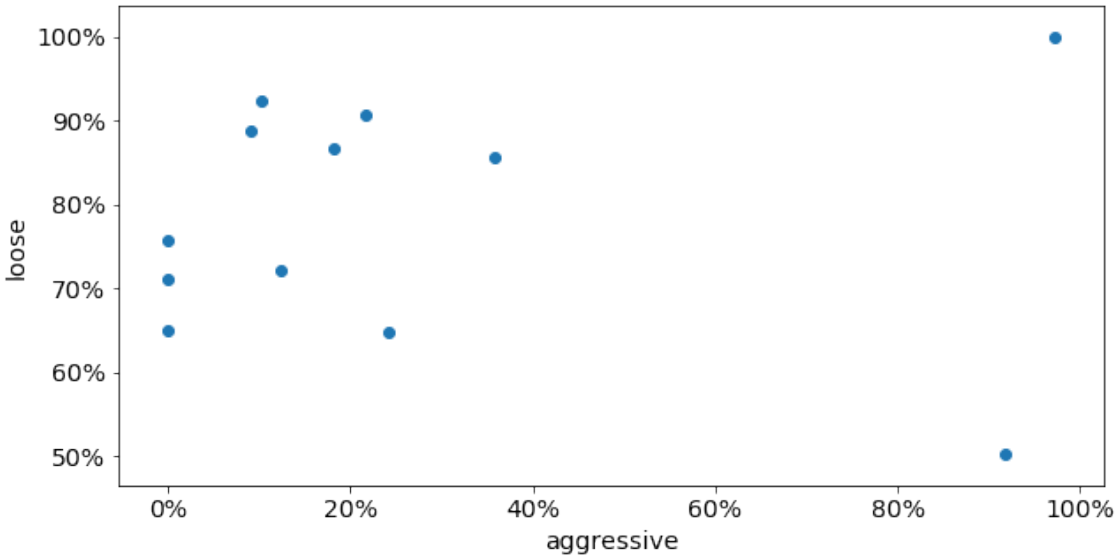

図 5.2: エージェントのプレイスタイル

表 5.3: ゲーム環境の構造

特徴量	次元数
現在のベット額	1
フォールドしていないプレイヤーの数	1
ラウンド開始時にフォールドしていなかったプレイヤー数	1
コミュニティカード	52
ホールカード	52

表 5.4: 順伝播型ニューラルネットワークの構造

レイヤー数	4
ノード数	107-64-64-3
活性化関数	ランプ関数（ReLU）
最適化アルゴリズム	SGD
損失関数	交差エントロピー

ラルネットワークを 4 つを学習器として、提案手法により分類と学習を行った。ベット額はスモールブラインドの額を 1 としている。テキサスホールデムのゲーム環境にはここで使用した 107 次元の特徴量以外にも、ポットや各プレイヤーの所持チップなどの情報が含まれる。しかし今回使用した 12 種類のエージェントが持つ行動ルールは、それらを使用しないため省略している。分類結果と予測精度は表 5.5 のようになった。学習器 1 から学習器 4 の列は各エージェントの行動を予測した際の予測精度を表し、最終的に各エージェントは、4 つの学習器の中で最も予測精度の高い太字のものに分類されている。ベースラインはそのエージェントの対戦ログのうち、最も選択回数の多い行動の割合を示し、常に同じ行動を予測したときに達成できる最大の精度である。学習器 1 に分類されたエージェント C と D、E は予測精度がベースラインより高く、入力となるゲーム環境の特徴を反映して予測できていることがわかる。同様にエージェント B と F も学習器 2 の予測精度もベースラインより高い。エージェント A は可能な限りベットまたはレイズを選択する、極端なアルゴリズムを持つが、これには学習器 4 が対応しており、予測精度はベースラインと同等である。残りのエージェントは全て学習器 3 に分類されており、これらの予測精度はベースライン以上とは言えないものの、概ね同等の精度が得られた。全体としてはどのエージェントも、分類された学習器によってベースラインと概ね同等かそれ以上の精度で予測されており、訓練データとかけ離れた戦略を学習した学習器はなかったと言える

表 5.5: 各エージェントに対する予測精度

エージェント	学習器 1	学習器 2	学習器 3	学習器 4	ベースライン
A	7.2%	32.0%	2.7%	**97.2%**	97.2%
B	10.5%	**81.7%**	3.8%	45.4%	50.8%
C	**74.0%**	30.7%	65.3%	0.0%	64.5%
D	**74.0%**	33.5%	63.4%	9.0%	62.7%
E	**77.6%**	31.5%	71.4%	0.0%	70.8%
F	44.8%	**81.8%**	48.2%	15.6%	48.9%
G	69.3%	53.1%	**74.5%**	0.0%	75.9%
H	62.5%	34.8%	**71.5%**	15.9%	71.1%
I	66.9%	47.5%	**79.8%**	7.8%	80.6%
J	52.2%	34.8%	**55.8%**	30.7%	54.8%
K	69.4%	42.4%	**80.8%**	9.6%	82.8%
L	63.0%	36.2%	**71.6%**	20.0%	70.7%

5.2.3 分類と学習結果の分析

エージェントの分類結果がプレイスタイルを反映しているかを調べるため、図 5.2 をクラスタ別に色分けしたものを図 5.3 に示す。クラスタ 1 とクラスタ 3 は、比較的近いプレイスタイルのエージェントで構成されていることがわかる。一方でクラスタ 2 に分類された 2 つのエージェントのプレイスタイルは、アグレッシブ度が大きく異なり近いとは言い難い。この結果については分類が上手く行われていないと考えることもできるが、学習器 2 がアグレッシブ度とルース度というプレイスタイルには表れない、2 つのエージェントの戦略の共通点を学習できた可能性もある。エージェント A は他のいずれのエージェントともプレイスタイルが大きく異なり、単独でクラスタ 4 を構成している。

次に学習器により学習された戦略のプレイスタイルを調査するため、ルールベースの 12 個のエージェントに 4 つの学習器を加えた、計 16 個のプレイヤーによって 5.2.1 節と同等の条件で対戦を行い、ログを記録した。各プレイヤーのスコアとプレイ傾向を表 5.6 示す。学習器 1 と 3、4 のスコアは、エージェント A を除くルールベースエージェントと比較して大きく劣っており、必ずしも訓練データとなったエージェントと同等の強さが得られるわけではないことがわかる。一方で学習器 2 は全エージェント中 3 番目の強さであり、クラスタ 2 のうちエージェント B よりは弱いものの、エージェント F より高いスコアを獲得している。また 5.2.1 節の対戦ではスコアが最も高かったエージェント A が、一転して最低のスコアを記録していることから、テキサスホールデムにおいては同じ戦略を持つプレイヤーでも、相手

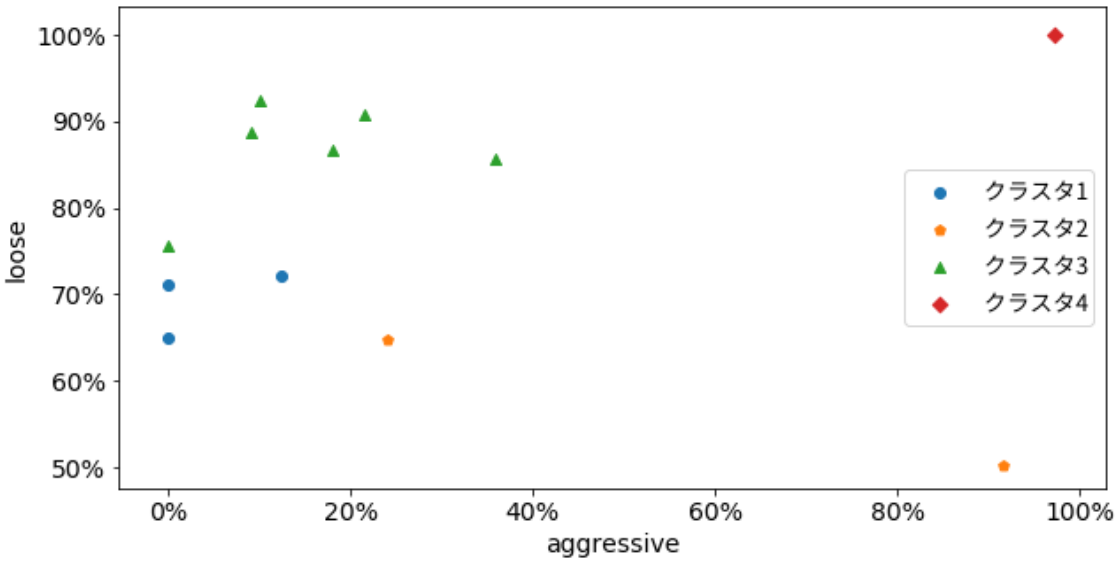

図 5.3: 分類結果とプレイスタイルの関係

の組み合わせによって大きくパフォーマンスが異なることが判明した。各学習器のスコアも他のエージェントの組み合わせによって、この実験結果とは大きく異なるスコアを出す可能性がある。

この対戦結果をプレイスタイルに基づいてプロットしたものを図 5.4 に示す。学習器 1 と学習器 2、学習器 3 のプレイスタイルは、訓練データとなったエージェントのプレイスタイルの平均にはなっておらず、極端にルースパッシブやタイトパッシブな戦略となっている。このことからルース度とアグレッシブ度という、テキサスホールデムにおける一般的な指標に基づいた評価では、提案手法によって生成した AI が、対戦ログに含まれる戦略に則ったものであると結論付けることはできない。

5.3 パラメータ依存のエージェントによる実験

5.2 節では、人の知識に基づいて記述されたルールを戦略として持つエージェントの分類を行い、プレイスタイルに基づく分析を行った。しかしプレイスタイルはあくまでも行動の統計であり、各エージェントの持つ戦略同士が類似しているかを、定量的に測定することは不可能であった。そこでパラメータに依存するルールを戦略とするエージェントを複数作成し、分類結果をパラメータに基づいて分析することで、分類結果と戦略の類似性の関係を明らかにするための実験を行った。

表 5.6: 各エージェントと学習された戦略の対戦結果

エージェント	スコア	Fold	Check+Call	Bet+Raise	ルース度	アグレッシブ度
A	-1	0	3499	45797	100.00%	92.90%
B	0.645	3624	331	3150	48.99%	90.49%
C	0.525	3539	6869	0	66.00%	0.00%
D	0.416	3426	12178	1265	79.69%	9.41%
E	0.354	3524	12975	0	78.64%	0.00%
F	0.442	3510	6753	919	68.61%	11.98%
G	0.313	3197	13792	0	81.18%	0.00%
H	-0.02	2557	20924	3099	90.38%	12.90%
I	0.312	2427	16376	1447	88.01%	8.12%
J	-0.013	2908	17139	7619	89.49%	30.77%
K	-0.272	1831	26635	2266	94.04%	7.84%
L	-0.15	2265	22594	4522	92.29%	16.68%
学習器 1	-0.553	437	37377	123	98.85%	0.33%
学習器 2	0.467	3413	4980	87	59.75%	1.72%
学習器 3	-0.592	135	39217	0	99.66%	0.00%
学習器 4	-0.875	0	3469	44513	100.00%	92.77%

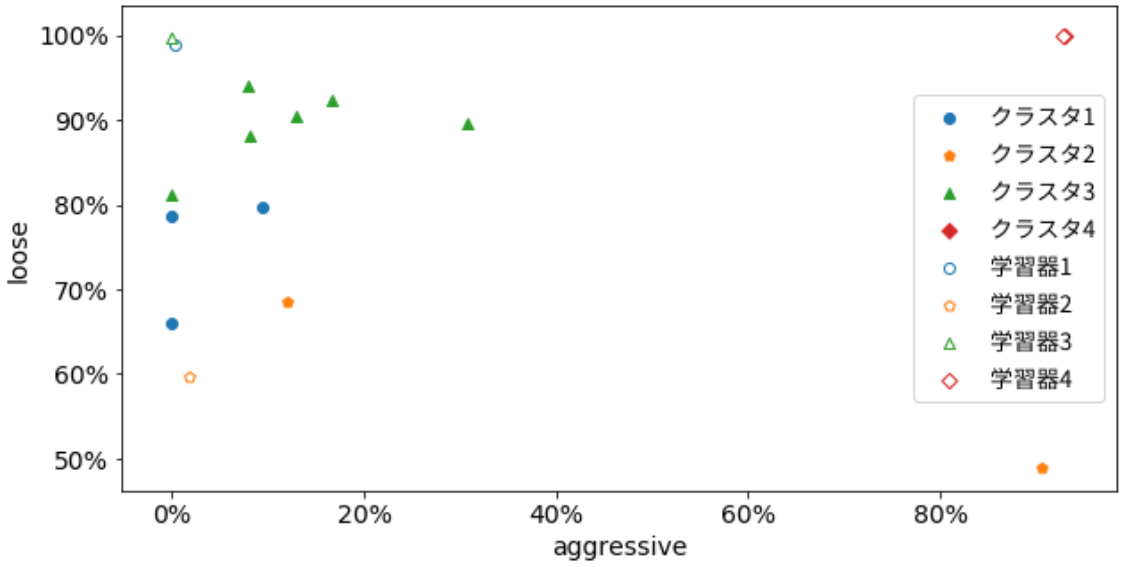

図 5.4: プレイスタイルに基づく分類の可視化

5.3.1 エージェントの作成

本実験の目的は提案手法の実用性を示すことではなく、戦略の類似性と分類結果の関係を明らかにすることであるため、複雑で多くの要因が意思決定に影響を及ぼすゲームでは、十分な測定結果が得られない可能性がある。そこでゲームの参加人数を 2 人に制限した上で、前半ラウンドは全てコールとチェックで進行し、後半ラウンドのデータのみを扱うこととした。本実験で使用するエージェントが従うルールの概要は以下の通りである。

1. 現在のホールカードとコミュニティカードから互いにフォールドしなかった場合の勝率を推定する。
2. 選択できる全ての行動に対して勝った場合と負けた場合の所持チップを推定する。
3. 各行動の期待値をパラメータに基づく補正を掛けて計算し、最も主観的な期待値の高い行動を選択する。

手順 1 の勝率の推定ではまず、ホールカードとコミュニティカードを合わせたときの最も強い役を判定する。ただし 3 ラウンド目ではコミュニティカードが 1 枚不足しているので、残りの 1 枚としてあり得る 46 パターン全てについて役の判定を行う。役の判定に続いて強さの数値化を行う。数値化の手順は 2.1.3 で示した役の弱い方から順に $0, 1, ..., 8$ を与えた上で、カードのランクにも弱い方から順に $0, 1, ..., 12$ を割り振り、優先順位の高い方からランクに対応する数値に、$1/13^1, 1/13^2, ...$ を掛けて加算する。例としてホールカードとコミュニティカードの和集合が、「♠7 ♠4 ♣7 ♢7 ♠A ♢2 ♡6」だった場合、最も強い役は「♠7 ♣7 ♢7 ♠A ♡6」のスリーカードである。スリーカードの強さは 2 であり、7、A、6 のランクの強さはそれぞれ 5、12、4 であるので、この役の強さは $2 + 5/13 + 12/13^2 + 4/13^3 \simeq 2.457$ となる。3 ラウンド目では強さの数値は 46 パターンの平均を取る。最終的な勝率の推定値は数値化した強さに応じて表 5.7 の値を使用する。

手順 2 の勝った場合と負けた場合の所持チップの推定では計算を簡易化するため、以降は全てコールかチェックでゲームが進行すると仮定して計算を行う。現在の所持チップを b、ポットと現在のラウンドでこれまでに賭けられた総額を m とすると、

表 5.7: 強さ毎の勝率の推定値

役の強さ	勝率
1 未満	0.1
1 以上 1.5 未満	0.3
1.5 以上 2 未満	0.5
2 以上 3 未満	0.7
3 以上 5 未満	0.9
5 以上	1

所持チップの推定値 b' は

$$b'_{\text{fold}} = b'_{\text{check,lose}} = b \tag{5.4}$$
$$b'_{\text{call,win}} = b'_{\text{check,win}} = b + m \tag{5.5}$$
$$b'_{\text{call,lose}} = b - 4 \tag{5.6}$$
$$b'_{\text{bet,win}} = b'_{\text{raise,win}} = b + m + 4 \tag{5.7}$$
$$b'_{\text{bet,lose}} = b'_{\text{raise,lose}} = b - 8 \tag{5.8}$$

となる。定数項 $-4, +4, -8$ はコールやベット、レイズの際に支払うチップの額を示している。

手順 3 では手順 1 と 2 で推定した勝率と所持チップを用いて期待値の計算を行うが、パラメータによって戦略を変化させるため、式 (3.1) の価値関数に推定された所持チップを与えてから主観的な期待値の計算を行う。実際の各行動の主観的な期待値 e は勝率を p とすると、

$$e_a = \begin{cases} v(b'_a) & (a = \text{fold}) \\ pv(b'_{a,\text{win}}) + (1-p)v(b'_{a,\text{lose}}) & (a \in \{\text{check, call, bet, raise}\}) \end{cases} \tag{5.9}$$

となる。エージェントは最終的に最も主観的な期待値の高い行動を選択する。

式 (3.1) のパラメータの内 α, β の 2 つを、それぞれ [0.5, 1] の範囲で 0.1 刻みに変化させ、このアルゴリズムに従うエージェントを計 36 個生成した。これらを同一エージェント同士を含む全ての組わせで対戦を行い、一つのエージェントにつき計 8192 個の、ゲーム環境と行動のペアからなる対戦ログを記録した。

5.3.2 戦略の分類結果と分析

生成した対戦ログのゲーム環境を表 5.8 に示す 56 次元の特徴量に変換し、表 5.9 の構成の 4 個の順伝播型ニューラルネットワークを学習器として、提案手法により

表 5.8: ゲーム環境の構造

特徴量	次元数
所持チップ	1
ポット	1
自分が現在のラウンドで賭けたチップ数	1
相手が現在のラウンドで賭けたチップ数	1
ホールカードとコミュニティカードの和集合	52

表 5.9: 順伝播型ニューラルネットワークの構造

レイヤー数	5
ノード数	56-64-64-64-3
活性化関数	ランプ関数（ReLU）
最適化アルゴリズム	SGD
損失関数	交差エントロピー

戦略の分類と学習を行った。特徴量のうち所持チップは初期の所持チップとの差を取った値であり負の値も取り得る。分類結果をエージェントの持つパラメータ α,β を軸としてプロットしたところ、図 5.5 のようになった。概ね α が大きく β が小さい領域がクラスタ 2、α が小さい領域がクラスタ 3、中間の領域がクラスタ 1 となっており、提案手法による分類はパラメータの傾向に基づいて、類似する戦略を同じクラスタに分類することができた。しかし、分類には 4 つの学習器を使用したものの、対戦ログが 1 つも属さないクラスタが発生したため、クラスタの数は実際には 3 つになった。

次に学習器が学習した戦略が、元となったエージェントの戦略を反映しているかを確かめるため、複数のゲーム環境を与えて出力される行動の割合を計測した。実験の手順として、ゲーム環境のうち所持チップとポット、両プレイヤーのベット額を固定し、ホールカードとコミュニティカードの計 7 枚をランダムに複数回生成して、これを入力としたときの各行動の出力頻度を計測する。まずクラスタ 3 に分類された対戦ログによって学習された学習器 3 に対する計測結果を表 5.10 に示す。ポットが 4 の行はポットと互いのベット額が最小値の場合の条件である。このとき学習器 3 は所持チップが小さいとレイズ、所持チップが大きいとコールを選択し、変化はおよそ-5 から-3 の間で起こっている。またポットが 36 の行はポットと互いのベット額が最大の場合の条件である。このときも最小値の場合と同様に、所持チップが-37 から-36 の間を境に行動がレイズからコールに変化している。学習器 3 の訓練データであるクラスタ 3 の対戦ログは、式 (3.1) の α の値が低いエージェントのものである。α の値が小さいと価値関数は x が正の領域での傾きの変化が急激であること

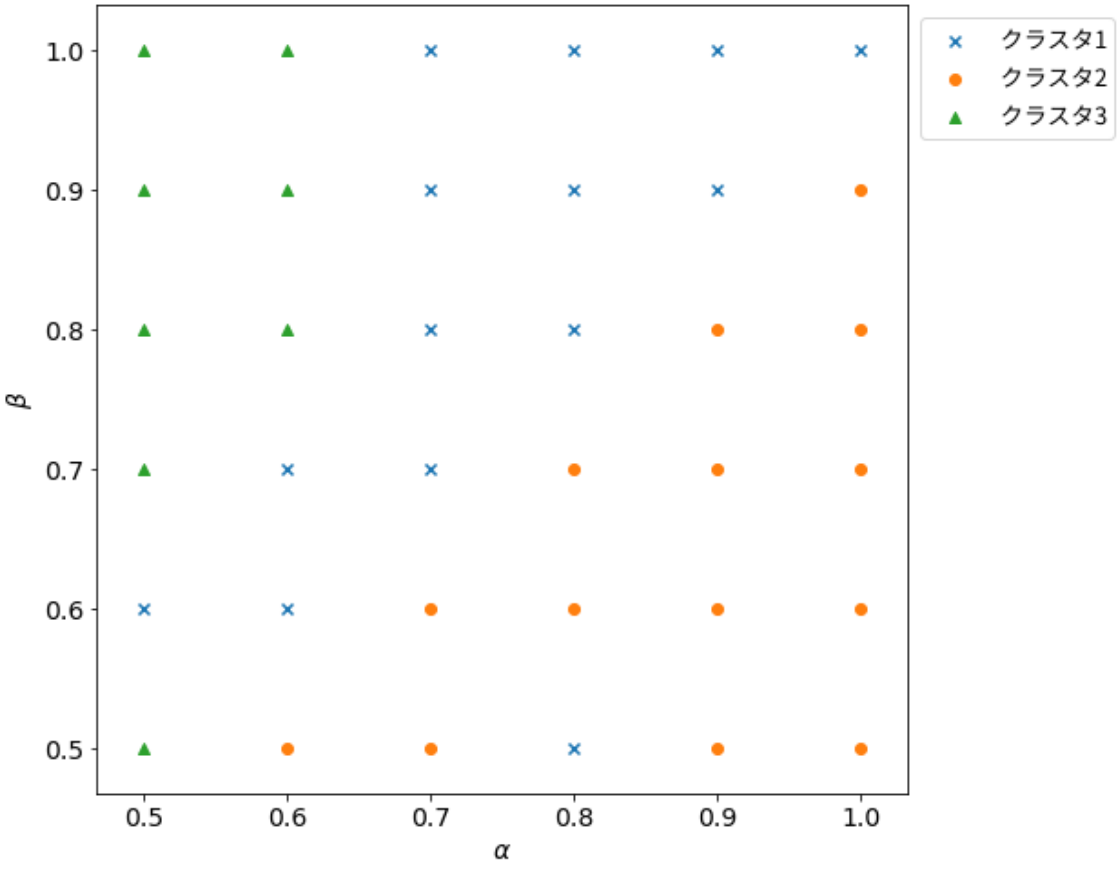

図 5.5: パラメータ α, β に基づく分類の可視化

表 5.10: 学習器 3 の戦略

所持チップ	ポット	自分のベット額	相手のベット額	Fold	Call	Raise
-10	4	0	4	0.0%	0.0%	100.0%
-6	4	0	4	0.0%	0.0%	100.0%
-5	4	0	4	0.0%	0.1%	99.9%
-4	4	0	4	0.0%	9.8%	90.2%
-3	4	0	4	0.0%	91.4%	8.6%
-2	4	0	4	0.0%	100.0%	0.0%
5	4	0	4	0.0%	100.0%	0.0%
-50	36	12	16	0.0%	0.0%	100.0%
-38	36	12	16	0.0%	0.0%	100.0%
-37	36	12	16	0.0%	1.1%	98.9%
-36	36	12	16	0.0%	99.8%	0.2%
-35	36	12	16	0.0%	100.0%	0.0%
0	36	12	16	0.0%	100.0%	0.0%

から、この価値関数をもつエージェントは、行動後の所持チップの期待値が正の場合に損失回避傾向が強くなる。よって学習器 3 の戦略もこのようなエージェントの戦略を学習した結果、所持チップが一定以上のときに、レイズよりローリスクローリターンなコールを選択するようになったと考えられる。また行動が変化する点の所持チップが 0 でないのは、フォールドを選択していないことから、コールやレイズの期待値は現在の所持チップより高い値であり、所持チップが 0 より小さいときにそれらの期待値が 0 になるからだと考えられる。

次に学習器 1 に対する同様の計測結果を表 5.11 に示す。学習器 3 のときと同様に所持チップが小さいときはレイズを選択するが、ポットと互いのベット額が最小値と場合と最大値の場合のどちらも、-2 から-1 の間を境にコールを選択するようになる。一方で学習器 3 との相違点として、所持チップが 0 から 1 の間で再びレイズを選択するようになることが挙げられる。ここで式 (3.1) 示した価値関数の $x > 0$ の範囲での導関数

$$v'(x) = \alpha x^{\alpha-1} \tag{5.10}$$

の概形を図 5.6 に示すと、$v'(x)$ は $x = 0$ 付近で急激に変化しその後は緩やかに変化していることがわかる。プロスペクト理論におけるリスク回避傾向の要因の一つは、$x > 0$ の範囲における価値関数の傾きの変化であり、x が 0 に近いほどリスク回避傾向が強まると考えられる。つまり学習器 1 の戦略は、訓練データであるクラスタ 1 に属するエージェントのパラメータ α が十分に低くなかったことから、コールやレイズの期待値が、リスク回避傾向の特に強まる 0 付近の正の値のときのみコールを

表 5.11: 学習器 1 の戦略

所持チップ	ポット	自分のベット額	相手のベット額	Fold	Call	Raise
-10	4	0	4	0.0%	0.0%	100.0%
-3	4	0	4	0.0%	0.0%	100.0%
-2	4	0	4	0.0%	22.9%	77.1%
-1	4	0	4	0.0%	97.5%	2.5%
0	4	0	4	0.0%	41.2%	58.8%
1	4	0	4	0.0%	0.0%	100.0%
10	4	0	4	0.0%	0.0%	100.0%
-10	36	12	16	0.0%	0.0%	100.0%
-3	36	12	16	0.0%	0.0%	100.0%
-2	36	12	16	0.0%	4.7%	95.3%
-1	36	12	16	0.0%	97.4%	2.6%
0	36	12	16	0.0%	100.0%	0.0%
1	36	12	16	0.0%	1.7%	98.3%
2	36	12	16	0.0%	0.0%	100.0%
10	36	12	16	0.0%	0.0%	100.0%

選択し、それを超えると再びレイズを選択するようになったと考えられる。

学習器 2 の戦略に対しても同じ手法で計測を行ったが、この戦略はこの実験で起こりうるほとんどのゲーム環境に対して、ベットかレイズを選択することが明らかになった。また、どの学習器の戦略もほとんどカードの情報を無視して行動を決定しており、各エージェントに対する行動予測の精度は最大でも 66.5%と高くなかった。特にフォールドはどの学習器においても一度も選択されなかった。この原因として、前半ラウンドはコールとチェックのみを行い、後半ラウンドのみ行動を選択するというゲーム設定では、フォールドの期待値が相対的に高くなることがあまりなく、対戦ログにフォールドを選択したログがあまり含まれていなかったことが挙げられる。実際に対戦ログの全ての行動の内フォールドの割合は 3.5%であり、学習器がフォールドを学習できなかった可能性が高い。

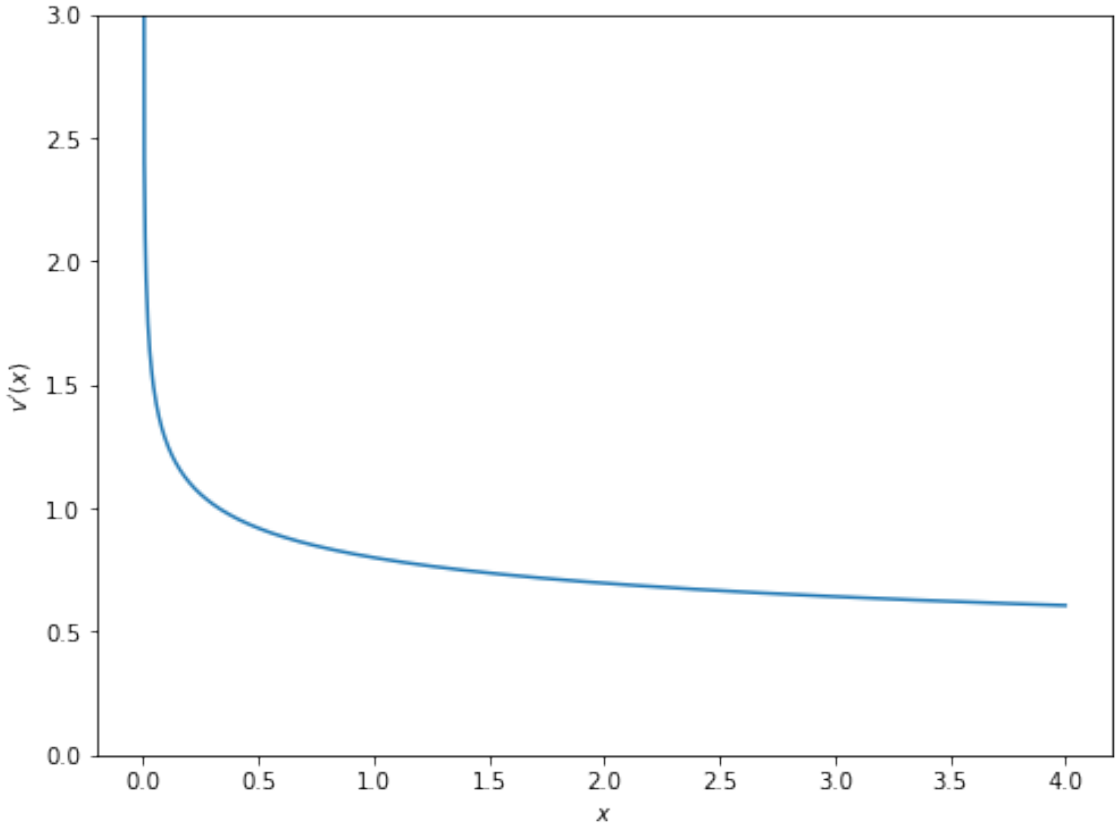

図 5.6: $\alpha = 0.8$ のときの価値関数の導関数

第6章　おわりに

6.1　まとめ

本研究では多様な戦略をもつゲーム AI の作成を行うため、対戦ログに基づいて類似の戦略を分類し、同時に順伝播型ニューラルネットワークなどの学習器によって、クラスタ毎の戦略を学習する手法を提案した。また提案手法の有効性を検証するため、分類と学習の対象として、パラメータを持つ単純な関数、人の知識に基づくルールベースのポーカーエージェント、パラメータに基づくルールベースのポーカーエージェントの 3 つで実験を行った。

単純な関数に対する実験では、提案手法はパラメータに沿って関数を適切に分類できることを示した。人の知識に基づくポーカーエージェントに対する実験でも、提案手法は概ねプレイスタイルに沿って戦略を 4 つのクラスタに分類することができた。しかし学習された戦略のプレイスタイルは、各クラスタの訓練データの平均値から大きく外れており、人間プレイヤーの戦略に則した AI を生成できているとは言えない結果となった。パラメータに基づくポーカーエージェントに対する実験でも、最初の実験と同様に概ねパラメータに沿って適切な分類を行うことができた。人の知識に基づくエージェントはそれぞれ全く異なるアルゴリズムを持つため、プレイスタイルの近さと戦略の類似性の関係は明確ではなかったが、この結果により提案手法は、類似する戦略を同じクラスタに分類する能力があることが確認できた。さらにこの実験に使用したエージェントは、プロスペクト理論におけるリスク愛好やリスク回避のバイアスを含んでおり、学習された戦略は訓練データに応じてリスク回避の傾向を持つことが明らかとなった。しかし学習された戦略の行動は偏りが多く、対戦ログの戦略に則した AI を生成するという目的に対しては改善の余地がある。

6.2　今後の課題

実験により提案手法は対戦ログに基づいて類似する戦略を分類することができるものの、各クラスタを代表するような戦略を適切に学習できていないことがわかった。これを改善する方法として、学習器として使う順伝播型ニューラルネットワー

クの構造の最適化や、学習器としてランダムフォレストやサポートベクターマシンなど、ニューラルネットワーク以外のモデルを適用することが考えられる。また戦略の分類は *k*-means 法をベースにしたアルゴリズムを使用しているため、*k*-means++ 法 [22] や *x*-means 法 [23] を参考に、学習器の初期化やクラスタ数の決定方法に関して改善できる可能性がある。

謝辞

本研究を進めるにあたり、長期に渡って知識面や研究の方向性についてご指導を賜りました橋山智訓准教授に、この場をお借りして心より感謝申し上げます。また、多くの助言を頂いただけでなく、実験への協力をしてくださった橋山研究室の方々や、研究室での活動において様々な面で支えてくださった田野俊一教授ならびに、田野研究室の皆様にも深く感謝申し上げます。

参考文献

[1] Murray Campbell, A.Joseph Hoane, and Feng hsiung Hsu. Deep blue. *Artificial Intelligence*, 134(1):57 – 83, 2002.

[2] 公益社団法人 日本将棋連盟. 第 2 回将棋電王戦／五番勝負, 2013. https://www.shogi.or.jp/match/denou/2/index.html.

[3] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

[4] 人狼知能プロジェクト. 第 4 回人狼知能大会, 2018. http://aiwolf.org/4th-aiwolf-contest.

[5] 電気通信大学. Uecda-2019 コンピューター大貧民大会, 2019. http://www.tnlab.inf.uec.ac.jp/daihinmin/2019/.

[6] A. I. Esparcia-Alcázar, A. Martínez-García, A. Mora, J. J. Merelo, and P. García-Sánchez. Controlling bots in a first person shooter game using genetic algorithms. In *IEEE Congress on Evolutionary Computation*, pages 1–8, July 2010.

[7] 片寄 晴弘 福嶋 良平. 2D アクションゲームにおける島モデル GA を用いた多様な振舞いの獲得. 情報処理学会論文誌, 58(11):1756–1764, nov 2017.

[8] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[9] Noam Brown and Tuomas Sandholm. Superhuman ai for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424, 2018.

[10] 24/7 Games LLC. Free poker games, 2018. https://www.247freepoker.com/.

[11] K. Burns. Style in poker. In *2006 IEEE Symposium on Computational Intelligence and Games*, pages 257–264, May 2006.

[12] C. Bauckhage, A. Drachen, and R. Sifa. Clustering game behavior data. *IEEE Transactions on Computational Intelligence and AI in Games*, 7(3):266–278, Sep. 2015.

[13] Drachen Anders, Alessandro Canossa, and Georgios N Yannakakis. Player modeling using self-organization in tomb raider: Underworld. In *2009 IEEE symposium on computational intelligence and games*, pages 1–8. IEEE, 2009.

[14] Tongfang Sun Baoxiang Wang nd and Xianjun Sam Zheng. Beyond winning and losing: Modeling human motivations and behaviors using inverse reinforcement learning. *CoRR*, abs/1807.00366, 2018.

[15] 上田陽平. 遺伝的アルゴリズムによる人間のレベルに適応する多様なオセロ AI の生成. 第 *27* 回ゲーム情報学研究会, *2012*, 2012.

[16] 鶴岡 慶雅 杉本 直樹. 戦略の動的推定による 2 人対戦ゲーム接待 AI の提案. In ゲームプログラミングワークショップ *2018* 論文集, volume 2018, pages 114–119, nov 2018.

[17] 佐々木 守 岩田 穆 小野 将寛, 汐崎 充. 強化学習を用いた対戦相手適応型戦略モデル. 電子情報通信学会技術研究報告. *NC,* ニューロコンピューティング, 103(228):61–66, jul 2003.

[18] Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–292, 1979.

[19] Amos Tversky and Daniel Kahneman. Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and uncertainty*, 5(4):297–323, 1992.

[20] Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisý, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, 356(6337):508–513, 2017.

[21] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, Berkeley, Calif., 1967. University of California Press.

[22] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. Technical Report 2006-13, Stanford InfoLab, June 2006.

[23] Dan Pelleg, Andrew W Moore, et al. X-means: Extending k-means with efficient estimation of the number of clusters. In *Icml*, volume 1, pages 727–734, 2000.