

大阪電気通信大学
総合情報学部 デジタルゲーム学科
2022 年度 卒業論文

機械学習で得られるポーカーの勝率と 確率論の比較検証

HW19A031 奥出 陸
デジタルアーキテクチャ研究室
指導教員：上善 恒雄

2023 年 2 月 1 日

概要

現状, 機械学習はその発達によりさまざまなところで利用されつつある. それは国家や企業だけでなく私たちの身の回りでも多く見られている. しかし, 機械学習の信用性は数字として実際に示されているものは少ない. 幅広く多くの人に機械学習の精度についての研究を行ってほしい.

本稿では, 機械学習の精度検証を行なってほしいため, 今私たちが簡単に利用できる GoogleDrive の Colaboratory で機械学習モデルを制作し, 機械学習を行い, そこから出力される数値比較をして機械学習の精度を検証する研究した. 研究方法としてはトランプゲームのテキサスホールデムポーカーのディーラーの手札 2 枚とプレイヤーの手札 2 枚と共有のカード 5 枚と勝敗を出力するプログラムを制作し, そのデータから, プレイヤーの手札 2 枚と共有のカード 5 枚と勝敗のデータのみ抽出して訓練データとする. 機械学習モデルに訓練データを入力し, 学習させ, テストデータから勝率を計算する. また, そのテストデータから実際に勝てるかもしれない勝率を計算し, 機械学習モデルから出力された勝率と実際に勝てるかもしれない勝率を比較する実験を行った.

研究結果としては, 関連研究には劣るがある程度の勝率を維持できる機械学習モデルが完成した. また, 機械学習モデルから出力された勝率は実際に勝てるかもしれない勝率を上回り, 機械学習は確率論の視点からみると精度が高く, 信用性が高いと分かった [1].

目次

1	はじめに	2
1.1	研究背景	2
1.2	研究の目的	2
1.3	機械学習と確率論	2
1.4	本論文の構成	2
2	機械学習の概要	2
2.1	教師あり学習	3
2.2	教師無し学習	3
2.3	強化学習	3
2.4	sklearn(scikit-learn)	3
2.5	不完全情報ゲーム	3
2.6	完全情報ゲーム	3
3	関連研究	4
3.1	ポーカーの統計データを用いたプレイヤーの実力指標	4
3.2	不完全情報ゲームにおける強化学習	6
3.3	完全情報ゲームにおける教師あり学習	7
4	ポーカーについて	8
4.1	ポーカーの種類とプレイ手順	8
4.2	ポーカーの役	11
5	実験内容	12
5.1	統計について	12
5.2	確率	12
5.3	使用言語について	12
5.4	環境	12
5.5	使用データについて	13
5.6	制作した機械学習	15
5.7	理論確率の計算方法	15
5.8	プレイデータの信頼度	17
5.9	実験手順	19
6	実験	19

6.1	実験結果	19
6.2	考察	19
7	まとめと今後の課題	20

1 はじめに

1.1 研究背景

今日機械学習の信用性は数字として実際に示されているものは少ないため、今回制作する機械学習モデルを通して数字として算出し、確率論と比較する事で機械学習の精度を明白にし信頼できるものであると確率論から証明したいと考え研究を始めた。

1.2 研究の目的

本研究の目的は、現在日常的に使用されている機械学習をテキサスホールデムポーカーを用いて調査し、確率論から比較分析した上で、機械学習の確率論の視点からの精度を検証することである。一般的に、機械学習には教師あり学習と教師なし学習と強化学習の3つの手法がある。それぞれの手法で活用できる範囲が異なる。

1.3 機械学習と確率論

機械学習とは機械に大量のデータを与え、それをもとに学習させることによって人間から要求される答えの正答率を高めるものである。その正答率を高めるために機械学習が用いる手法として統計、確率、傾向を用いている。観測されたデータをもとに、統計や確率の理論を使って model の未知パラメータを予測して定めるものである。このことから機械学習と確率論は切り離すことができない非常に密接な関係にあると言える。

1.4 本論文の構成

本論文では、2章で機械学習の概要と使用する機械学習の種類と完全、不完全情報ゲームの説明と今回使用した環境について説明を行う。3章では関連研究について触れる。4章では今回用いるポーカーの種類と大まかなルール説明などを説明する。5章では実験で用いるデータや理論確率の計算方法、実験手順について説明する。6章では5章の実験手順に則って実験を行い、そこから得た結果から考察をする。7章では、まとめと今後の課題を述べた。[2]。

2 機械学習の概要

機械学習とは入力データから継続的に繰り返し学習をすることで、その中から特定のパターンを探し出すことである。また、学習を繰り返す事で変化を予測し、その結果からできる限り正しい精度を高める事である [3]。

2.1 教師あり学習

入力データが存在していて、正解も与えられている事で統計的に回帰や分類をしていく手法である（例：システム不正の検出、メールのスパム判定）[4].

2.2 教師無し学習

入力データは存在しているが、正解が与えられない事で入力データのパターンやグループ分けをしていく手法である（例：自動運転）.

2.3 強化学習

最初からデータが存在するわけではなく、システム自身が試行しながら報酬が与えられる事でどの行動が最善なのかを見つけ出す手法である（例：ロボットの歩行、将棋）.

2.4 sklearn(scikit-learn)

Python で使用できるオープンソースプロジェクトのライブラリ、またはフレームワークと呼ばれるものである。また機械学習全般のアルゴリズムが実装されたライブラリである。統計学、パターン認識、データ解析などの多岐にわたる技法が使用できる。特に分類、回帰、クラスタリング、次元削減をも目的として多く使用されている。

2.5 不完全情報ゲーム

不完全情報ゲームは動学ゲームであるが、少なくとも 1 人のプレイヤーが何かしらの意思決定を行う少なくとも 1 つの時点において、それより以前に行われた少なくとも 1 人のプレイヤーが行った何かしらの意思決定の内容を観察、確認できないゲームのことである（例：ポーカー、麻雀など）[5].

2.6 完全情報ゲーム

完全情報ゲームは動学ゲームであり、さらにプレイヤーが何かしらの意志決定を行うときに、そのプレイヤーより前に意志決定を行ったプレイヤーの何かしらの意志決定の内容を観察、確認できるゲームのことである（例：チェス、将棋など）[6].

3 関連研究

3.1 ポーカーの統計データを用いたプレイヤーの実力指標

ノーリミットホールデムポーカーでのプレイヤーの実力を公平に指標する目的で、統計データにもとづく機械学習の研究が行われていた。この研究で行われた研究方法は、オンラインポーカーサイト「pokerstars」の 50,000 個のプレイデータからさまざまな複雑な要素で統計を出し、それをもとに要素ごとの特徴量を学習させ、実際にプレイした内容が正解データと近ければ近いほどプレイヤーの実力指標が高く出力されるものである。

以下の表は統計として使われた要素を示したものである。

統計	内容
VPIP	参加率
Limp	リンプでの参加率
PFR	レイズでの参加率
Cold Call	レイズに対してのコールで参加する確率
3bet In Position	インポジションで 3 ベットする確率
3bet Out of Position	アウトオブポジションで 3 ベットする確率
Fold to 3Bet IP	インポジションで 3 ベットに対してフォールドする確率
Fold to 3Bet OOP	アウトポジションで 3 ベットに対してフォールドする確率
4Bet IP	インポジションで 4 ベットする確率
4Bet OOP	アウトポジションで 4 ベットする確率
Fold to 4Bet	4 ベットに対してフォールドする確率
5Bet(Raise 4Bet)	5 ベットする確率
WTSD	フロップを見たうえでショーダウンまで行く確率
WSWSF	フロップを見たときの勝率
WSSD	ショーダウンでの勝率
Agg	$(\text{ベット数} + \text{レイズ数}) \div (\text{コール数})$
River Call Efficiency	$(\text{リバーでコールして勝った額}) / (\text{リバーでコールした額})$
Flop Cbet IP	フロップインポジションでの C ベット率
Flop Cbet OOP	フロップアウトポジションでの C ベット率
Flop Fold vs Cbet OOP	フロップアウトオブポジションで C ベットに対してフォールドする確率

表 1 統計の要素:1

統計	内容
Flop Raise Cbet OOP	フロップアウトオブポジションで C ベットに対してレイズする確率
Flop Fold vs Cbet IP	フロップインオブポジションで C ベットに対してフォールドする確率
Flop Raise Cbet IP	フロップインオブポジションで C ベットに対してレイズする確率
Bet Turn OOP vs Missed Flop Cbet OOP	アウトオブポジションでフロップ C ベットされなかった時にターンでベットする確率
Turn Cbet IP	ターンインポジションでの C ベット確率
Turn Cbet OOP	ターンアウトオブポジションで C ベット確率
Bet Flop vs Missed CBet	フロップで C ベットされなかったときに、ベットする確率
Fold Turn IP Skip Cbet	インポジションでフロップ C ベットせずに、ターンのベットに対してフォールドする確率
Delayed Turn CBet OOP	アウトオブポジションでフロップ C ベットせず、ターンでベットする確率
Check Fold Flop As PFR	プリフロップをレイズで参加し、アウトオブポジションのフロップでチェックフォールドする確率
Check Raise Flop As PFR	プリフロップでレイズ参加し、アウトオブポジションのフロップでチェックレイズする確率
Delayed Turn CBet IP	インポジションでフロップ C ベットせず、ターンでベットする確率
Turn Fold vs CBet OOP	アウトオブポジションでターンの C ベットに対してフォールドする確率
Bet River OOP vs Missed Turn CBet IP	インポジションでターン C ベットされなかったとき、リバーでベットする確率
Turn Fold vs CBet IP	インポジションでターンの C ベットに対してフォールドする確率
Skip Turn CB abd Fold To River Bet IP	アウトオブポジションでターン C ベットをせずに、リバーのベットに対してフォールドする確率
Check Fold Turn OOP Skip CBet	アウトオブポジションでターン C ベットせずに、チェックフォールドする確率
Check Raise Turn OOP Skip CBet	アウトオブポジションでターン C ベットせずに、チェックレイズする確率

表 2 統計の要素:2

統計	内容
Bet Turn IP vs Missed CBet	インポジションでターン C ベットされなかったときに、ベットする確率
Flop CBet-Fold IP TOT	インポジションでフロップ C ベットをし、レイズに対してフォールドする確率
Flop CBet-Fold OOP TOT	アウトオブポジションでフロップ C ベットをし、レイズに対してフォールドする確率
River CBet OOP	リバーアウトオブポジションでの C ベット確率
River CBet IP	リバーインポジションでの C ベット確率

表 3 統計の要素:3

大平 翼, 横山 昌平, ポーカーの統計を用いたプレイヤーの実力評価 [6] より引用し一部改変

表 1 の要素は実際に用いられた要素の中からいくつかを抜粋したものである。この研究ではノーリミットホールデムポーカーのルールが非常に複雑かつ一般的でないものが多く含まれていて、実際に用いられた要素も複雑かつ多量にある。

本研究では統計として用いるデータの要素を 2 枚のプレイヤーのカードの組み合わせと 3 枚の確認できる共有のカードの組み合わせと実際の勝敗の 3 要素に絞って学習させて研究を行う。

3.2 不完全情報ゲームにおける強化学習

CFR (Counterfactual Regret Minimization) が利用できないような状態空間の広い多人数ゲームに対しても NFSP (Neural Fictitious Self-Play) が利用できることを示す示す目的で、テキサスホールデムポーカーを題材に機械学習の強化学習の手法を用いて研究が行われていた。この研究で行われた研究方法は、一般的なポーカーのルールよりもはるかに簡略化されたポーカーであるクーン・ポーカー (Kuhn Poker) にテキサスホールデムのルールを少し織り交ぜたルールで強化学習させ、そこから可搾取量を求め、それからテキサスホールデムポーカーにおける多人数不完全情報ゲームのナッシュ均衡を求めるものである。

以下は 2 人でのクーンポーカーにおける可搾取量とイテレーション回数を図にしたものである。

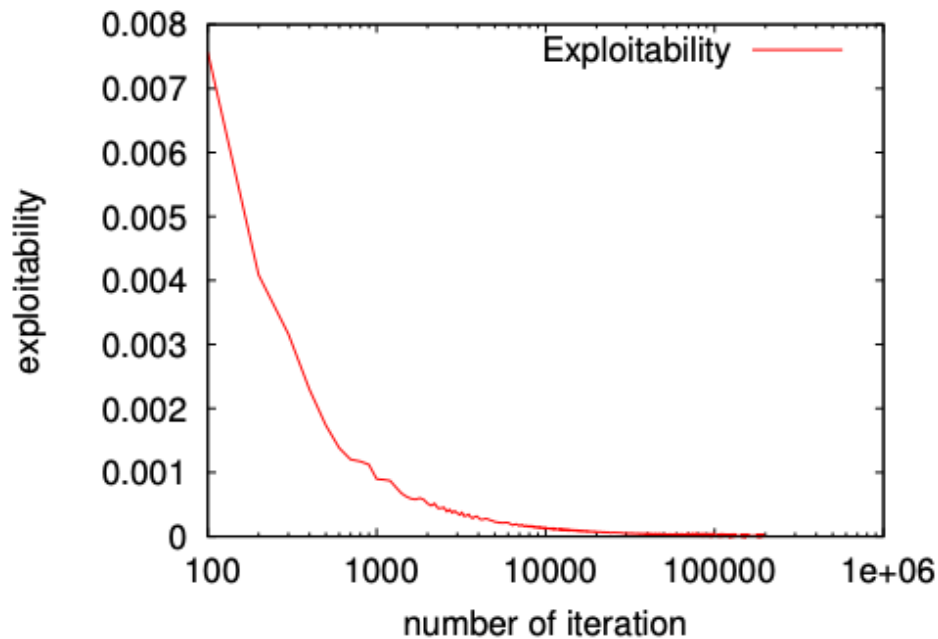


図1 プレイヤーが2人のときのクーンポーカーにおける可搾取量
河村 圭悟, 水上 直紀, 鶴岡 慶雅, 多人数不完全情報ゲームにおける仮想自己対戦を用いた強化学習, 2016 年 10 月 28 日, 194.[7] より引用

図1は縦軸が可搾取量を示し、横軸がイテレーション回数を示したものである。この図1からイテレーション回数を重ねるごとに可搾取量が減少していき、イテレーション回数が10,000回を超えたあたりから限りなく0に近づいている。よって強化学習を用いた多人数不完全情報ゲームの機械学習は成功している。

本研究ではこちらの研究のように強化学習ではなく、分類型の教師あり学習を用いて研究を行う。

3.3 完全情報ゲームにおける教師あり学習

棋譜の解説や初心者の指導といった実用と、コンピュータを通じて人間の思考方法をより深く理解することの実現を目的として、棋譜の指手を教師とした学習を行っており、また2つの状況を比較するのではなく、それぞれの状況について未来を予測しながら比較研究したものである [8]。

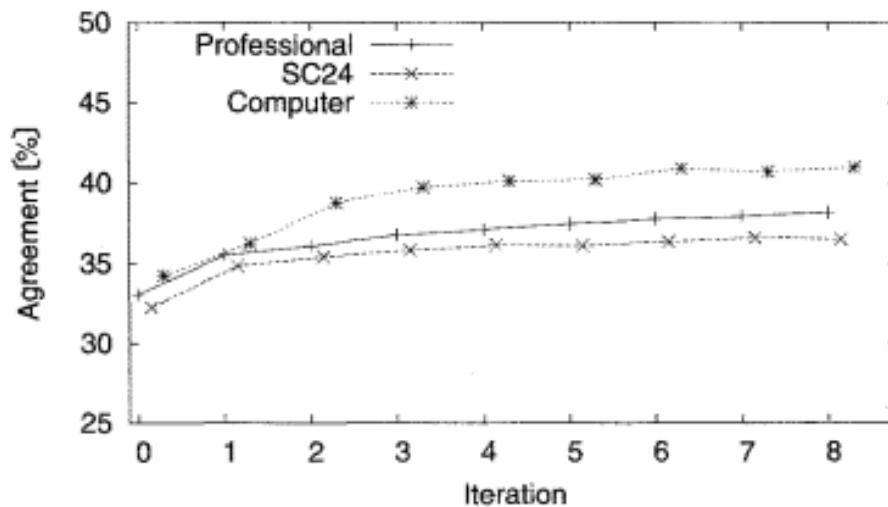


図2 それぞれプレイ時の正答率

金子 知適, コンピュータ将棋の評価関数と棋譜を教師とした機械学習 (Evaluation Functions of Computer Shogi Programs and Supervised Learning Using Game Records), 2012 年 1 月 1 日, 80[8] より引用

図2はこの研究内で将棋をプレイした時の棋譜にもとづいたプロフェッショナル, アマチュア, 学習後コンピューターのそれぞれの一致率がプロフェッショナルが約 38%, アマチュアが約 36%, 学習後コンピューターが約 42% と示しているものである. これによって完全情報ゲームにおいての教師あり学習ができたと解釈できる.

この研究では将棋という完全情報ゲームであるのに対し, 本研究ではテキサスホールデムポーカーという不完全情報ゲームを用いて研究を行う.

4 ポーカーについて

4.1 ポーカーの種類とプレイ手順

ポーカーは世界で有名なトランプゲームの1つである. ポーカーにはドローポーカー・スタッドポーカー・ホールデム・チャイニーズポーカーなどさまざまな種類があるが, 今回は日本ではあまり馴染みがないが世界的に最も流通している「テキサスホールデムポーカー」を使用していく [9].

最初に本来のテキサスホールデムポーカーのルールを説明する.

1. カードを伏せて参加者全員にカードが 2 枚ずつ配られる.
2. プレイヤーはその 2 枚のカードを見て, 他のプレイヤーに勝てると思ったら, チップを賭けることができるし, 勝てないと思ったら降りることもできる. 順番に全員の判断を決定していき, 全員の

賭け金が揃ったら、ディーラーの前に表向きにカードが3枚配られる。

3. 各プレイヤーはその場に出された3枚と、自分の持っている2枚のカードを合わせ、前回と同じように賭けるか降りるか判断する。
4. 全員の賭け金が揃ったら、ディーラーが表向きにカードを1枚配る。場に配られている4枚のカードと、自分の持っている2枚のカードを使って、前回までと同じように賭ける、もしくは降りる。
5. 全員の賭け金が揃ったら、ディーラーが表向きにカードを1枚配る。この場に配られた5枚のカードと自分が持っている2枚のカードで、賭けを行う。前回までと同じように賭け、もしくは降りる。
6. 最終的に残っているプレイヤー同士で場の5枚、手持ちの2枚のカードの合計7枚のうち、5枚でポーカーの役を作り、誰が強いかで勝敗が決定する。勝ったプレイヤーが、今までの賭けで貯まったチップを獲得する。途中で自分以外のプレイヤーが全て降りてしまった場合も残ったプレイヤーが勝ちとなる。

前述した通り公式にはこういったルールになっているが、今回はディーラーとの1対1を構想しているので掛金やレイズなどの難しいルールを省くこととする。

1. カードを伏せて参加者全員にカードが2枚ずつ配られる。
2. プレイヤーはその2枚のカードを見る。その後ディーラーの前に表向きにカードが3枚配られる。
3. プレイヤーはその場に出された3枚と、自分の持っている2枚のカードを合わせ役(手札)を作り勝負する。
6. 最終的にプレイヤーとディーラーで場の5枚とそれぞれの手持ちの2枚のカードの合計7枚のうち、5枚でポーカーの役を作り、どちらが強いかで勝敗が決定する。今回は引き分けの場合はディーラーの勝ちとする。

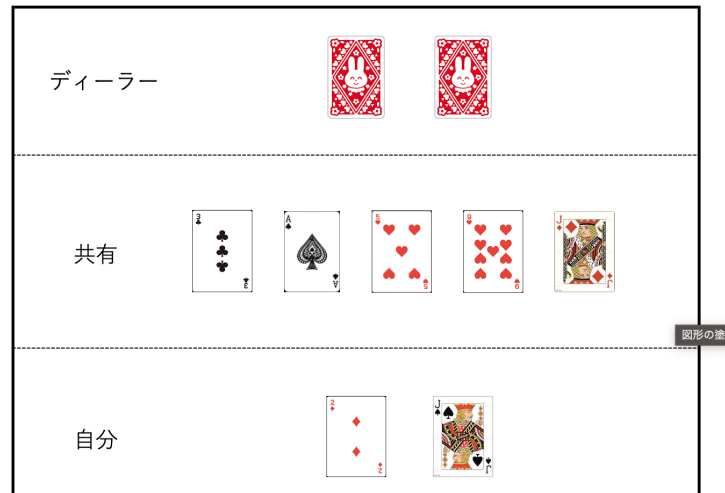


図 3 第 3 者視点から見た場の様子

図 3 は第 3 者視点での場の様子を表したものである. ディーラーの 2 枚の手札は確認することができず, 共有部の場の 5 枚とプレイヤーの 2 枚の手札のみ確認できるものとする.



図 4 プレイヤー視点から見た

図 4 はゲーム中のプレイヤー視点の場の様子を表したものである. ゲームの最初から最後までディーラーの手札が見えない状態でスタートし, 勝敗を決定する時に公開する. また, 共有部の場の 5 枚のうち 3 枚しか見えていない状態であることとする. この状態でプレイヤーの 2 枚の手札と, 共有部の場の 5 枚のうち確認することができる 3 枚の 5 枚でプレイヤーの 5 枚の手札を予測して勝負

する.

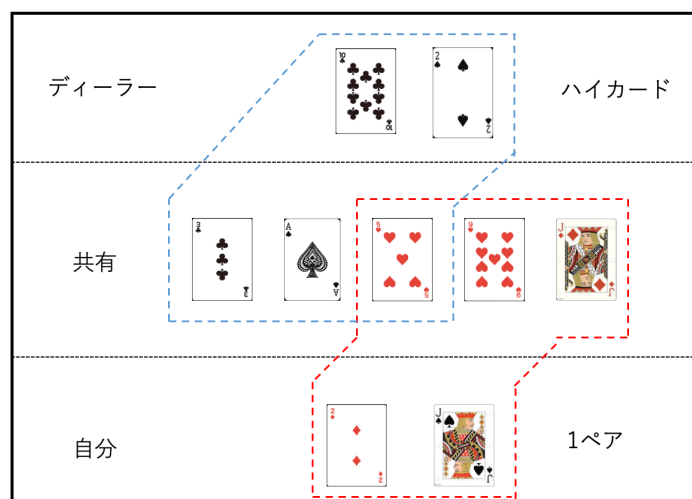


図 5 ゲーム終了時の場の様子

図 5 はゲームの勝敗を決定する際の全てのプレイヤー視点の場の様子を表したものである. 図 5 の青点線部で囲まれたものがディーラーの最終の 5 枚の役 (手札) となる. また, 図 5 の赤点線部で囲まれたものがプレイヤーの最終の 5 枚の役 (手札) である. 図 3 の場合, ディーラー側の役はハイカードで, 最も弱い役である. また, プレイヤー側の役は J (ジャック) のワンペアとなる. 今回の場合, ディーラーがハイカードで, プレイヤーはワンペアとなり, プレイヤーの作った役の方が強い役となるのでプレイヤーの勝利となる.

4.2 ポーカーの役

ポーカーの役について説明する. また, 今回はトランプのマークでは役の強弱はないものとする. ポーカーにはロイヤルストレートフラッシュ, ストレートフラッシュ, フォーカード, フルハウス, フラッシュ, ストレート, スリーカード, ツーペア, ワンペア, ハイカードの 10 種類の役がある [10].

ロイヤルストレートフラッシュはマークが全て同じの「エース (A)」, 「キング (K)」, 「クイーン (Q)」, 「ジャック (J)」, 「10」の役のことである.

ストレートフラッシュはマークが全て同じの 5 枚の連続した数字のカードで作られた役のことである.

フォーカードは同じ数字のカードが 4 枚手札にある場合に有効な役のことである.

フルハウスは同じ数字のカードが3枚と、その数字とは異なる数字で同じ数字のカードが2枚手札にある場合に有効な役のことである。

フラッシュはマークが全て同じであるカード5枚が手札にある場合に有効な役のことである。

ストレートはマークを無視して、連続した数字のカードが5枚手札にある場合に有効な役のことである。

スリーカードは同じ数字のカードが3枚手札にある場合に有効な役のことである。

ツーペアは同じ数字のカードが2枚と、さらにその数字と異なる数字で同じ数字のカードが2枚手札にある場合に有効な役のことである。

ワンペアは同じ数字のカードが2枚手札にある場合に有効な役のことである。

ハイカードは手札の全てのカードに同じ数字のカードもなく、5枚の手札のマークが異なっている場合に役無しと言われる役のことである。

5 実験内容

5.1 統計について

本研究ではポーカーにおける、不完全情報の状況下における勝敗を統計として使用する。統計の例としては、プレイヤーの2枚の手札の数字、マークの組み合わせ、また確認できる共有部の場の3枚のカードなどが挙げられる。これらの統計はプレイヤーのポーカーの勝敗に大きく影響していると考えられる。これらの統計はプレイヤーごとに大きく異なるものではなく、誰にでもポーカーの勝率を上昇させることに利用できる要素だと考える。これらの統計を機械学習を使う上での特徴量として利用することで、ポーカーで長期的に勝てるかを判別する分類 model を構築する。

5.2 確率

本稿ではポーカーのプレイデータを用いて統計的に機械学習を行い、そこから得られる勝率を統計確率とする。また、ポーカーのプレイデータから確率論的に得られる勝率を理論確率とする。

5.3 使用言語について

本稿は機械学習 (Machine Learning) を用いるため、広く普及していて他のライブラリよりも機械学習の実装をより簡単に試行することができるため、プログラミング言語は Python である。

また、他のライブラリよりも機械学習の実装をより簡単に試行することができるため、拡張モジュールとして「Numpy」, 「Pandas」, 「sklearn(Scikit-learn)」を用いた [11]。

5.4 環境

MacOS(13.1)

Python(3.9)

Numpy(1.21.6)

Pandas(1.3.5)

sklearn(1.0.2)

5.5 使用データについて

本研究では入力層に 1,000,000 個ものポーカーのプレイデータが必要であるため, それらのデータを出力するプログラムを制作した. このプログラムではプログラム言語として Python を使用し, モジュールとしては「Numpy」, 「Pandas」を用いた. また Pandas の DataFrame を用いてデータの表にし, csv にした [12].

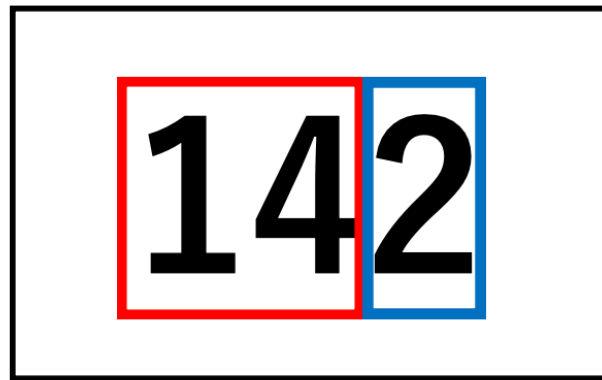


図6 トランプのカードの数字とマーク

図6は今回トランプのカードの数字とマークを合わせて表したものである. 図6のように142という数字が出力された場合は青枠で囲われた142の1の位の数字がマークを示し, 赤枠で囲われた部分がカードの数字を示している. また, 今回においてはカードの数字の1(A)は14と表記され, マークについては1: ハート, 2: スペード, 3: ダイヤ, 4: クローバーとして出力される.

プレイ回数 (回目)	プレイヤーの手札	共有のカード	プレイヤーの役	勝敗
0	[74, 134]	[41, 133, 33, 61, 23]	[13, 13, 7, 6, 4]	[1]
1	[81, 92]	[21, 83, 42, 52, 63]	[8, 8, 9, 6, 5]	[0]
2	[32, 93]	[84, 82, 72, 24, 142]	[8, 8, 14, 9, 3]	[0]
3	[122, 62]	[104, 112, 111, 53, 51]	[2, 11, 5, 12, 6]	[0]
4	[31, 124]	[91, 114, 43, 94, 64]	[3, 9, 12, 11, 9]	[0]
5	[23, 104]	[84, 52, 111, 143, 102]	[2, 10, 14, 11, 10]	[1]
6	[83, 62]	[141, 93, 64, 81, 132]	[2, 8, 6, 14, 13]	[1]
7	[33, 112]	[82, 103, 21, 124, 114]	[3, 11, 12, 10, 8]	[1]
8	[22, 121]	[42, 31, 94, 73, 71]	[2, 7, 12, 9, 7]	[0]
9	[74, 24]	[101, 134, 43, 34, 142]	[14, 13, 10, 7, 4]	[0]
10	[93, 121]	[43, 62, 34, 44, 113]	[6, 4, 12, 11, 9]	[0]
11	[82, 73]	[102, 134, 51, 141, 22]	[14, 13, 10, 8, 7]	[0]
12	[21, 41]	[71, 33, 123, 124, 31]	[2, 12, 3, 7, 4]	[0]
13	[103, 53]	[111, 64, 32, 74, 144]	[14, 11, 10, 7, 6]	[0]
14	[143, 122]	[23, 94, 104, 72, 101]	[10, 10, 14, 12, 9]	[0]
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
999985	[62, 131]	[42, 81, 94, 73, 54]	[4, 9, 6, 13, 8]	[1]
999986	[134, 113]	[41, 122, 21, 133, 73]	[13, 13, 12, 11, 7]	[1]
999987	[64, 33]	[74, 101, 62, 43, 94]	[3, 6, 10, 9, 7]	[0]
999988	[123, 124]	[51, 82, 144, 61, 63]	[12, 2, 12, 6, 14]	[1]
999989	[72, 141]	[121, 24, 132, 42, 111]	[14, 13, 12, 11, 7]	[1]
999990	[54, 131]	[34, 112, 31, 44, 114]	[2, 11, 3, 13, 5]	[1]
999991	[144, 53]	[22, 41, 63, 44, 73]	[5, 4, 14, 7, 6]	[0]
999992	[103, 83]	[122, 24, 141, 31, 111]	[14, 12, 11, 10, 8]	[0]
999993	[82, 123]	[113, 74, 64, 43, 71]	[7, 7, 12, 11, 8]	[0]
999994	[124, 131]	[133, 112, 121, 94, 61]	[12, 13, 12, 11, 13]	[1]
999995	[21, 142]	[93, 132, 84, 54, 134]	[2, 13, 14, 9, 13]	[0]
999996	[141, 61]	[54, 53, 144, 82, 113]	[14, 6, 14, 5, 5]	[1]
999997	[83, 74]	[123, 21, 73, 92, 34]	[7, 7, 12, 9, 8]	[1]
999998	[33, 101]	[102, 24, 103, 111, 143]	[10, 3, 10, 14, 10]	[1]
999999	[32, 84]	[22, 63, 93, 133, 71]	[3, 13, 9, 8, 7,]	[0]

表 4 ポーカーのプレイデータ

表 4 は実際に出力させた 1,000,000 個のポーカーのプレイデータの DataFrame を表にしたものである.

5.6 制作した機械学習

本研究では大量のデータを使用し, 統計を用いるので教師あり学習で分類型を用いることにした. プログラミング言語として Python 用い, 機械学習の中核としては「sklearn」を使用した. また, 分類型に広く使われている sklearn の「RandomForestClassifier」を用いた. RandomForestClassifier は機械学習のアルゴリズムの 1 つで, 学習用のデータをランダムにサンプリングして多数の決定木を作成し, 作成した決定木をもとに特徴量の差から多数決で結果を決める手法で, 精度や汎用性が高く扱いやすい分析手法である.

本稿ではポーカーのプレイデータを入力し, 機械学習を行って学習した成果を用いてを行えるか得た統計勝率を理論確率と比較する研究した.

5.7 理論確率の計算方法

プレイヤーが確認できる 5 枚のカードからできる役を出力するプログラムを制作した. 以下のよう出力される.

- 9: ロイヤルストレートフラッシュ
- 8: ストレートフラッシュ
- 7: フォーカード
- 6: フルハウス
- 5: フラッシュ
- 4: ストレート
- 3: スリーカード
- 2: ツーペアー
- 1: ワンペアー
- 0: ハイカード

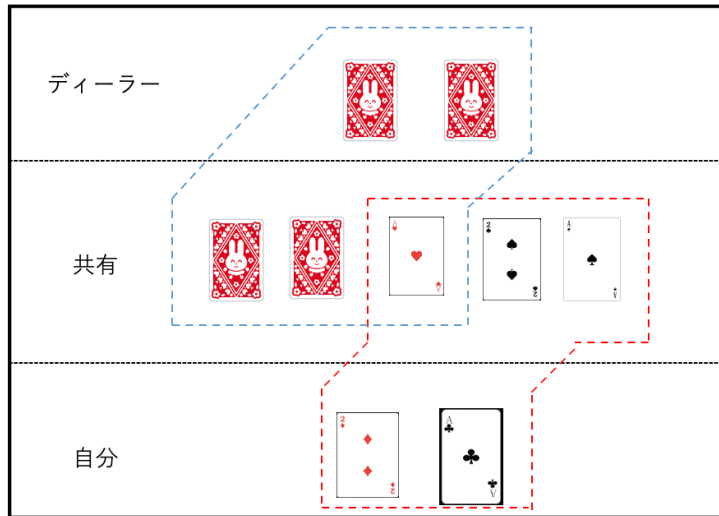


図 7 例として挙げる場の様子

図 7 を例に計算式を列挙する. 共有部の右から 3 枚が確認できるカードである.
 確認できる 5 枚のカードでできる役を数値として出力するプログラムから出力されるのが「6: フルハウス」である. フルハウスに勝つことのできる役はフォーカードまたはストレートフラッシュまたはロイヤルストレートフラッシュである.

(理論確率) を P とする。 $P = 1 - \left(\frac{4!}{47C_4} + \frac{4!}{47C_4} + \frac{4! * 11}{47C_4} + \frac{4P_2 * 45P_2}{47C_4} \right) = 1 - 24072/178365 = 0.865040...$

よって図 7 での理論確率は約 0.865 となる.

このように出力によって 6: フルハウス出力された場合の一般式はフルハウスの役の数字を n, m (フルハウスの役の 2 枚であるほうを $n, 3$ 枚である方を m) とする.

$n \geq 10$ の場合

$$P = 1 - \left(\frac{4!}{47C_4} + \frac{(15 - n) * 4!}{47C_4} + \frac{4! * 11}{47C_4} + \frac{4P_2 * 45P_2}{47C_4} \right)$$

$n \leq 6$ で, $m \geq 10$ の場合

$$P = 1 - \left(\frac{4!}{47C_4} + \frac{(n - 1) * 4!}{47C_4} + \frac{4! * 11}{47C_4} + \frac{4P_2 * 45P_2}{47C_4} \right)$$

$n \leq 6$ で, $m < 10$ の場合

$$P = 1 - \left(\frac{(n - 1) * 4!}{47C_4} + \frac{4! * 11}{47C_4} + \frac{4P_2 * 45P_2}{47C_4} \right)$$

10 > n > 6 で, m ≥ 10 の場合

$$P = 1 - \left(\frac{4!}{47C_4} + \frac{5!}{47C_4} + \frac{4! * 11}{47C_4} + \frac{4P_2 * 45P_2}{47C_4} \right)$$

10 > n > 6 で, m < 10 の場合

$$P = 1 - \left(\frac{5!}{47C_4} + \frac{4! * 11}{47C_4} + \frac{4P_2 * 45P_2}{47C_4} \right)$$

となる. 確認できる 5 枚のカードで他の役が出力されても同様に一般式を用いて計算できる.

5.8 プレイデータの信頼度

以下に実際に出力されたテキサスホールデムポーカーの役の出現回数をまとめた表を示す.

役名	出現回数(回)
9:ロイヤルストレートフラッシュ	0
8:ストレートフラッシュ	283
7:フォーカード	1724
6:フルハウス	26014
5:フラッシュ	30245
4:ストレート	46398
3:スリーカード	48356
2:ツーペアー	174135
1:ワンペアー	437890
0:ハイカード	174135

図 8 それぞれの役の出現回数

図 8 は 3.2.6 で説明した確認できる 5 枚のカードでできる役を数値として出力するプログラムから出力されたそれぞれの数値の回数をまとめたものである.

また, 以下にテキサスホールデムポーカーでのそれぞれの役が作られる確率を表にしたものを示す [13].

役名	出現確率(%)
9:ロイヤルストレートフラッシュ	0.003232062
8:ストレートフラッシュ	0.027850748
7:フォーカード	0.168067227
6:フルハウス	2.596102271
5:フラッシュ	3.025494123
4:ストレート	4.619382087
3:スリーカード	4.829869755
2:ツーペアー	23.49553641
1:ワンペアー	43.82254574
0:ハイカード	17.41191958

図9 テキサスホールデムポーカーでの役の出現確率

図9から比較すると出現回数と出現確率はほぼ同値を取っている. よって 1,000,000 個のテキサスホールデムポーカーのプレイデータには過度な偏りがなく, 信頼できるデータであると言える. 以下に2枚のプレイヤーの手札の組み合わせにおいて勝敗を色を区別して表したものを示す.

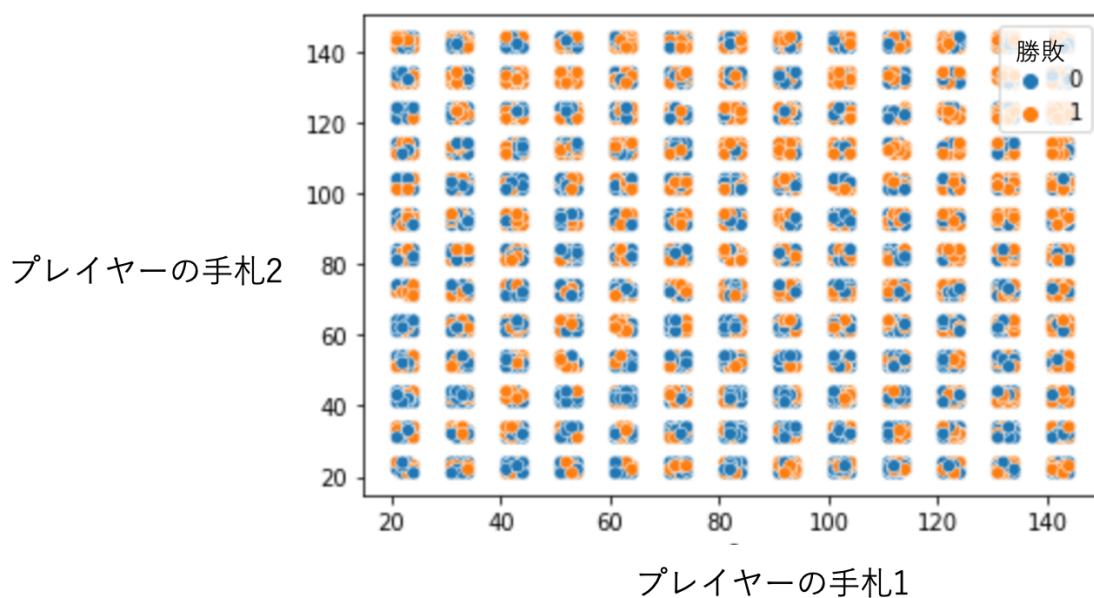


図10 プレイヤーの手札の数字とその勝敗

図10では縦横それぞれがプレイヤーの手札の数字となっていて, その手札のときの勝敗をオレンジを勝利, ブルーを敗北としている. この図10では右上にいくにつれて2枚のプレイヤーの手

札の組み合わせが強くなっている。それにあわせて色も左下から右上になるにつれてオレンジが多くなっているのがわかる。よって2枚のプレイヤーの手札の組み合わせは勝敗に大きく関係していると言える。

5.9 実験手順

1. 1,000,000 個のテキサスホールデムポーカーのプレイデータを入力層に入力する。
2. 学習を行い、分類 model の作成をする。
3. 作成した分類 model を用いて、テスト用のデータをテストさせる。
4. 作成した分類 model の統計確率を計算する。
5. テスト用のテキサスホールデムポーカーのプレイデータから理論確率を計算する。

6 実験

6.1 実験結果

ポーカーのプレイデータから機械学習を行い、学習前と学習後の精度の違いを比較するために実験を行った。

5.8 から確率論から見た学習の偏りもないと推測できる。

この 1,000,000 個のポーカーのプレイデータを学習 model に入力し、sklearn を用いて学習させた。また、本研究での学習用とテスト用のデータの比率は一般的な範疇内としそれぞれを 8:2 とした。

そこから得られた統計確率は 0.645765 となった。また、3.2.6 で説明した確認できる 5 枚のカードでできる役から確率を計算する一般式を利用して計算し、合算して平均を取り、得られた理論確率は 0.509749 となった。

これから理論確率の数値よりも統計確率の数値が約 0.135 上回っていた。

6.2 考察

実験から統計確率の方が理論確率よりも約 0.135 上回っていたことから機械学習の確率論の視点からの精度は高く、また確率論よりも秀でていることが分かった。

ただし、本研究では教師あり学習の分類 model を使用しており、大きく範囲を限定していることから機械学習としての大きな意味での信用性が得られてはいない。

また、6.1 で得た理論確率と統計確率に関してはテスト用データの勝利した回数の合算から導き出したものである。

本研究では不完全情報ゲームであるテキサスホールデムポーカーを用いたが、情報が不完全であるがために、分類 model の学習が完璧からは少し離れたところにあったと言える。

3.1 で述べた関連研究とは統計として用いるデータの要素の総数が大きく異なっている。本研究

では3つの要素から実験を行ったが、使用したテキサスホールデムポーカーのデータの総数が本研究は3.1で述べた関連研究の40倍もの量であることから近い実験結果が得られたと考える。

3.2で述べた関連研究では、機械学習に使用したデータの詳細が明記されていなかったので、図1から比較すると3.2で述べた関連研究では可採取量がイテレーション回数が10,000回を超えたあたりから限りなく0に近づいているのでクーンポーカーの強化学習は正答率がかなり高いことが分かる。勝率についても明記されていないが、本研究と比較するとやはり3.2で述べた関連研究のほうが精度がかなり高いと言える。今回扱ったポーカーのルールが若干異なるとはいえ、不完全情報ゲームであるポーカーにおいては強化学習の方が適性があると考ええる。

3.3で述べた関連研究では完全情報ゲームである将棋を扱っているが、棋譜の指手を教師としたとき、一致率は正答率と言い換えることができる。そこで3.3で述べた関連研究と比較したときに本研究の方が精度が高いと言える。これは将棋には駒を動かす手は約80~115であり、また将棋では取った駒を再度使うことができ、1回の勝負でかかる平均ターンは約115であり、1回の指手で80通りあるので棋士が指せる手は80~115であるので、これは実質無限であると解釈できる。よって教師あり学習には将棋は不向きであると言える。そこからポーカーは不完全情報ゲームであり、本研究では1度のプレイで9枚のカードを使用するので、52C9という膨大な数になるのでポーカーは強化学習のほうが適性が高いと言える。また、完全情報ゲームであっても将棋でいう指手の数が膨大な数であれば強化学習のほうが適性が高いと言える[14]。

実験結果と関連研究との比較から、機械学習の種類別の適性はあるものの大まかに言えば確率論の視点からみた機械学習は精度が高く、信用性が高いと考える。

7 まとめと今後の課題

本研究では不完全情報ゲームであるテキサスホールデムポーカーの大量のプレイデータから、分類 model を用いて教師あり学習をさせて、その統計確率と確率論の視点からみた、そのときのテキサスホールデムポーカーのプレイデータをもとに計算して求めた理論確率を比較する実験を行った。そこで得た結果から機械学習は確率論の視点からみると精度が高く、信用性が高いとであると分かった。しかし、精度を高めたり、機械学習をよりよいものへするには使用するデータや条件、状況によって機械学習の種類別の適性を考えて model を構築していく必要があることが分かった。

今後の課題としては本実験で使用したテキサスホールデムポーカーのプレイデータにプレイヤーの2枚の手札と3枚の共有部分のカード以外にもさまざまな要素を追加して特徴量の選択を増やしていったり、さまざまな機械学習の種類や model で試行するべき必要がある。

また、本実験ではディーラーとの1対1を想定して実験を行ったが複数人での同様の実験も行う必要がある。

参考文献

- [1] 杉山 将: “機械学習研究のこれまでとこれから” <https://linedevday.linecorp.com/2020/ja/sessions/6613/>, 2023 年 1 月 16 日.
- [2] “sklearn.ensemble.RandomForestClassifier” <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>, 2023 年 1 月 18 日.
- [3] “NRI 用語解説 — 技術 — 機械学習” https://www.nri.com/jp/knowledge/glossary/1st/ka/machine_learning, 2023 年 1 月 11 日.
- [4] “教師なし学習とは — 教師あり学習や強化学習との違い・活用事例・代表的なアルゴリズムを紹介” <https://ledge.ai/unsupervised/>, 2023 年 1 月 18 日.
- [5] “完全情報ゲームから不完全情報ゲームへ” <http://entcog.c.ooco.jp/entcog/event/20120318/matsubara.pdf>, 2023 年 1 月 23 日.
- [6] 大平 翼, 横山 昌平: “ポーカーの統計を用いたプレイヤーの実力評価” <https://proceedings-of-deim.github.io/DEIM2021/papers/E13-1.pdf>, 2021 年 3 月 3 日.
- [7] 河村 圭吾, 水上 直紀, 鶴岡 慶雅: “多人数不完全情報ゲームにおける仮想自己対戦を用いた強化学習” <http://altmetrics.ceek.jp/article/id.nii.ac.jp/1001/00175326/>, 2016 年 10 月 28 日.
- [8] 金子 知適: “コンピュータ将棋の評価関数と棋請を教師とした機械学習 (Evaluation Functions of Computer Shogi Programs and Supervised Learning Using Game Records)” https://www.jstage.jst.go.jp/article/jjsai/27/1/27_75/_pdf/-char/ja, 2012 年 1 月 1 日.
- [9] “ポーカーの種類とルール” <https://www.h-eba.com/heba/gamble/poker.html>, 2023 年 1 月 26 日.
- [10] “PokerLab TOKYO 【ポーカー役一覧】 テキサスホールデムポーカーの役を強さ順に解説” <https://poker-lab.jp/hands-ranking/>, 2023 年 1 月 27 日.
- [11] “Python 標準ライブラリ” <https://docs.python.org/ja/3/library/index.html>, 2023 年 1 月 12 日.
- [12] “python でテキサスホールデムのシミュレーション 1” <http://by-oneself.com/poker-python-1/>, 2023 年 1 月 15 日.
- [13] “カジノの王様 【保存版】 テキサスホールデムの確率まとめ | 勝率を高める最も重要なオッズ” <https://casino-kingdom.com/texas-holdem-probability/>, 2023 年 1 月 22 日.
- [14] “PHP オンライン衆知「プロ棋士が人工知能に負ける日」を 1996 年に予言した羽生善治氏” <https://shuchi.php.co.jp/article/6686>, 2023 年 1 月 27 日.