

コンピュータ大貧民における LSTM を用いた手札推定

神田 直樹^{1,a)} 伊藤 毅志^{1,b)}

概要: 近年, コンピュータ大貧民において, モンテカルロ法を用いたプログラムが上位を占めている. モンテカルロ法においては, プレイアウトの精度を高めることでプログラムの強化を行うことが知られている. プレイアウトの精度を高める方法の一つとして, プレイアウトの前に行う相手手札の推定をより正確にすることが挙げられる. 本研究では, 場の状況とその時のプレイヤーの行動という 2 つの時系列履歴から LSTM を用いることで, 推定率の向上を目指した. 評価実験の結果, 提案手法の手札推定法は先行研究の手札推定法に比べて, 中盤以降で精度が向上することを示した.

Estimation of Hand Using LSTM in Computer Daihinmin

NAOKI KANDA^{1,a)} TAKESHI ITO^{1,b)}

Abstract: In recent years, programs using the Monte Carlo method has occupied the top positions in the Computer Daihinmin (UEC-da). In the Monte Carlo method, it is known to strengthen the program by improving playout accuracy. One way to increase the accuracy of playout is to make the estimation of opponent hands done before playout more accurate. In this research, we tried to improve the estimation rate by using LSTM from the two time series histories of the situation of the field and the action of the player at that time. As the result of the evaluation experiment, the hand estimation method of the proposed method shows that the accuracy improves after the middle game compared to the hand estimation method of the previous research.

1. はじめに

1.1 背景

完全情報ゲームにおける人工知能研究はめざましく, 多くのゲームで人間トップを上回る段階に到達している. チェスは, 1997 年に IBM のディープブルーが当時の世界チャンピオンのカスパロフ氏に勝ち越した. 将棋も, 2015 年にはコンピュータ将棋が人間のトップの実力に追いつていることが客観的データから示され [1], 2017 年に行われた第 2 期電王戦では, Ponanza が現役の佐藤天彦名人に 2 連勝し, 上述の内容を裏付ける結果を示した [2]. 二人完全情報確定ゼロ和ゲームの中で, 最も開発が難しいとされてきた囲碁においても, Google 傘下の DeepMind 社が DNN (Deep Neural Network) の手法を用いて開発した

AlphaGO [3] が, 2017 年に世界ランク 1 位の柯潔九段との対局で 3 連勝した [4]. 完全情報ゲームの世界では, 既に AI は人間のトップを超えるレベルに到達したと出される. 二人完全情報確定ゼロ和ゲームの世界では, 既に AI は人間のトップを超えるレベルに到達したと出されるだろう. 一方, 不完全情報ゲームの人工知能の研究では, ポーカーや麻雀, 人狼などの研究が行われている [5][6][7]. ポーカーでは, 1 対 1 で掛け金に限度のあるルールにおいては, 人間を上回るプレイをする AI が現れているが, 多人数で掛け金に限度のないゲームにおいては, まだ人間のトッププレイヤーレベルに到達していない. このように, 不確定で多人数のゲームにおいては, まだ残された課題がある. その研究対象の一つとして, 日本で非常に多くのプレイヤーがいる大貧民というトランプゲームがある. その研究対象の一つとして, 日本で非常に多くのプレイヤーがいる大貧民というトランプゲームがある. 電気通信大学では, 2006 年から大貧民の AI 同士の大会である UECda [8]

¹ 電気通信大学
The University of Electro-Communications, Chofu, Tokyo
18-8585, Japan

a) kanda@minerva.cs.uec.ac.jp

b) ito@cs.uec.ac.jp

が開催され、研究対象としての地位を築いてきた。2009年には、乱数によるシミュレーションを行うことで行動を決定するモンテカルロ法を用いた fumiya[9] が優勝し、それ以降の優勝プログラムはすべてモンテカルロ法が用いられており、主流の手法になっている。コンピュータ大貧民におけるモンテカルロ法では、ある局面において他プレイヤーの手札を推定し、乱数を用いたシミュレーションを終局まで行う。(これをプレイアウトと呼ぶ。)モンテカルロ法を利用してより正しい解を得るためには、プレイアウトの質を向上させる必要がある。プレイアウトの質を向上させるための一手法として、他プレイヤーの手札をより正確に推定することが求められる [10]。他プレイヤーの手札推定の精度が上がれば、他プレイヤーの着手可能な手の予測も正確になる。これによって、モンテカルロ法におけるプレイアウトの質を高めることが可能になる。このように、多人数不完全情報ゲームにおいて相手の手をなるべく正確に予測することは、重要である。

1.2 目的

多人数不完全情報ゲームにおいて、相手の手の推定率を高めることは重要な意味を持つ。そのため、大貧民でも同様の研究が行われてきた。他プレイヤーの手札を推定する手法として、柳澤らは相手プレイヤーの行動履歴を用いた手札推定法を提案した [11]。この手法では、推定対象プレイヤーの行動履歴を分類し、分類に応じたカードの出現確率を用いて手札推定を行う。しかし、この手法では、「場の状況に応じた行動を考慮していない点」や「過去の行動履歴を正しい順序で認識できていない点」などの問題点がある。柳澤らの手法は何のカードを提出したかという行動のみに着目し、場の状況に応じた行動を考慮してないため、場に出ているカードや縛りが発生しているか等の場の状況が異なる場合も、全て同じ行動を行ったと認識してしまう問題がある。また、過去の行動履歴を正しい順序で認識できていないため、何のカードをどのターンで出したかというターン毎の提出カードの情報が正しい順序で記録されていない。本研究では、場の状況を考慮した行動をターン毎に正しい時系列順に保存する手札推定の手法を構築することで、より正確な他プレイヤーの手札推定の実現を試みる。具体的には、コンピュータ大貧民における場の状況とその時のプレイヤーの行動の2つの時系列履歴を用いた相手手札推定法を提案し、手札推定率の向上を目指す。場の状況とプレイヤー行動の2つのデータを時系列順に学習させるために、本研究では多くの特徴量を長期的な時系列データとして扱うことが出来る LSTM を利用する。また、時系列データを扱うことの効果を明らかにするために、比較実験として、時系列履歴を利用しないニューラルネットワークのみ(以降、NNとする)の実験も行った。

2. コンピュータ大貧民

コンピュータ大貧民は、大貧民をコンピュータ上でプレイさせるゲームである。コンピュータ大貧民の仕組みとしては、手札カード配布や順位による手札カード交換、後述の標準ルールに則ったゲーム進行と管理を自動で行うサーバーが用意されており、5つのクライアントプログラムはTCP/IPで接続し、対戦を行う。

電気通信大学においてコンピュータ大貧民の大会であるUECDaが2006年から開催されている。2009年の優勝プログラムとして乱数によるシミュレーションを行うことで行動を決定するモンテカルロ法を用いた fumiya[9]、2010年の優勝プログラムとして fumiya を改良した snowl[12]、2011年の優勝プログラムとして snowl に差分学習を適用させた crow[13] など、2009年以降の優勝プログラムにはモンテカルロ法が用いられる様になった。

3. 先行研究

3.1 コンピュータ大貧民における手札推定の有効性の検討

コンピュータ大貧民における手札推定の有効性の検討を行うべく、吉原らによる手札推定率と勝率の関係性を検証した研究がある [10]。吉原らは、snowl のプレイアウト中の手札推定の成功率を調整し、手札推定の成功率と試合で得られる得点にどのような変化が生じるか検証を行った。検証実験として、何も変更を加えていない通常 snowl を3つ、プレイアウト中の手札生成を全て乱数によって行う snowlr と手札推定の成功率を調整出来る改造版 snowl の5つで100000回対戦を行い、成功率毎の手札推定調整を行った snowl の得点の集計を行った。手札推定の成功率毎の手札配布は、サーバーから他プレイヤー全てのカード情報を受け取り、設定された推定率通りに手札配布を行う。

実験結果から、推定率が上がるにつれて平均得点が増加していく事が確認出来た。そして、完全推定出来ているものが平均点が一番高く、同じ推定率なら弱い手札から推定出来る方が平均点は良い事が確認出来た。弱い手札から推定した方が強い手札から推定するよりも得点が良い理由として、大貧民プレイのゲーム木を考えた際に弱いカードを推定する方が広く枝を展開することが出来るからであると筆者は述べている。

この結果から、手札推定率を上げる事はモンテカルロ法プレイヤーの強さの向上に繋がるという事が考えられる。

3.2 コンピュータ大貧民における手札推定手法

コンピュータ大貧民における手札推定の手法として、柳澤らは相手プレイヤーの提出手役履歴を用いた手札推定法を提案した [11]。柳澤らは、残り枚数が少ない場合の手札推定が重要であると考えた。そこで、柳澤らは、残り手札枚

数2~5枚の対象プレイヤーに対してカードの出現確率を用いた手札推定を行う推定手法を提案した。この手法は、対象の行動履歴を考慮しつつ手札推定を行えるため、推定対象のプレイヤー毎の分類に応じた出現確率を用いて推定を行うことが出来る。

しかし、先行研究の手法には2つの問題点がある。まず、異なる状況下でも同じ行動を行ったと認識してしまう点である。柳澤らの手法は何のカードを提出したかという行動だけに注目している。そのため、場に出ているカードや縛りが発生しているか等の場の状況が異なる場合にでも、全て同じ行動を行ったと認識してしまう問題がある。ランク2のカードを出す場合、場にカードが置かれていない時やある程度カードが場に置かれている時、また革命時や縛りが発生している時など、場の状況によってランク2のカードを出す意味は異なる。しかし、柳澤らの手法はこれら全ての行動を同じ行動として認識してしまう問題がある。次に、ターン毎の提出カードの情報が正しい順序で記録されていない点である。柳澤らの手法は推定時のターンまでに場に出したカードの数や種類だけに注目している。そのため、どのターンに何のカードが出されたか一切記録されない。手札に強いランクが多く、自分の順番を何回か回したいと考えて序盤にランク2のシングルを出した行動や、手札が少なく、残り手札で上がりを考えて終盤にランク2のシングルを出した行動のように、ターンによってカードを出す行動の意味合いは異なる。しかし、柳澤らの手法はこれらの行動を単なるシングル2を出した行動としてカウントしてしまう。

上記の問題点を解決するべく、本研究では以下の項目を満たす必要があると考えられる。

- 場に出ているカードや縛りが発生しているかどうかという場の状況を考慮した行動の認識
- ターン毎の提出カードを正しい時系列順に保存

そこで本研究では、場の状況を考慮した詳細な行動を特徴量とし、ターン毎の行動履歴を時系列データとして学習する事が出来るLSTMを用いる。

4. Long Short Term Memory(LSTM)

Long Short Term Memory(LSTM)は、RNNにて長期的な依存関係を学習する事が出来るように拡張したものである[15]。NNと異なる点は、RNNの中間層をLSTM-Blockに置き換えている。LSTMBlockには、Constant Error Carousel(CEC)と呼ばれるメモリと入力ゲート(Input-Gate)、忘却ゲート(ForgetGate)、出力ゲート(OutputGate)の三種類のゲートがある。

時刻 t において、 x_t を入力データ、 h_t を出力、 i_t 、 f_t 、 o_t をそれぞれ入力ゲート、忘却ゲート、出力ゲート、 c_t はCECとすると、LSTMは以下の式で表すことが出来る。

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f) \quad (2)$$

$$c_t = i_t \odot \tanh(W_{cx}x_t + W_{ch}h_{t-1} + b_c) + f_t \odot c_{t-1} \quad (3)$$

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + W_{oc}c_t + b_o) \quad (4)$$

$$h_t = o_t \odot \tanh(c_t) \quad (5)$$

式中の σ はシグモイド関数、 \odot はベクトルの要素ごとの積、 W は重み、 b はバイアス項である。

5. 提案手法

5.1 提案手法の概要

本研究では、場の状況とその時のプレイヤーの行動の2つの時系列履歴を利用したLSTMによる相手手札推定手法を提案する。提案手法の概要図を図1に示す。

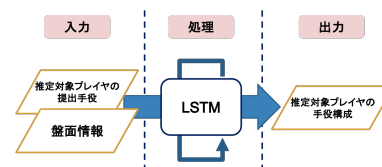


図1 提案手法の概要図

本手法は、相手手札を推定する際に、次のような手順を踏む。

- (1) 推定対象プレイヤーのターン時の盤面データを取得する。
- (2) 盤面データから、場の状況と推定対象プレイヤーの提出手役を特徴量とし、LSTMの入力層に与える。
- (3) 2で入力データを与えた際に出力された値を用いて、推定対象プレイヤーの手役構成を生成する。

5.2 入力層に使用する特徴量

異なる状況下での行動は同じ行動ではないと認識するためには、場の状況を考慮した行動を認識させる必要がある。そのため、入力層で用いる特徴量は場の状況と推定対象プレイヤーの行動を事細かに認識させる特徴量である必要がある。そこで、本研究では地曳らの提案した提出手役関数[16]にて用いている特徴量を参考に、推定対象プレイヤーの状態、場の状況、周りのプレイヤーの状態、提出カード、提出前と後による変化等の情報を入力層で用いる特徴量とした。特徴量の各項目は、真の場合に1、偽の場合に0をとる。作成した特徴量を以下の表1に示す。

5.3 提案手法で使用するLSTM

提案手法で使用するLSTMとして、Chainer(バージョン2.0.0)を用いた。LSTMの入力層のユニット数は表1の評価項目個数である205個であり、出力層のユニット数は各ランクの単体役・複数役・階段役に合わせて51項目用意

表 1 入力層に用いる特徴量

評価基準	要素数
推定対象プレイヤーの現在の所有枚数	12 通り
推定対象プレイヤーの階級	5 通り
未使用カード	53 通り
場のカード	53 通り
勝ち抜けたプレイヤーの数 (0,1,2,3)	4 通り
パスをしたプレイヤーの数 (0,1,2,3,4)	5 通り
出す前は革命かどうか	2 通り
出す前は縛りかどうか	2 通り
提出カード	53 通り
出した役の数 (0,1,2,3,4 枚以上)	5 通り
出した役の種類 (パス、単体役、複数役、階段役)	4 通り
革命を起こしたかどうか	2 通り
縛りを起こしたかどうか	2 通り
8 切りを起こしたか	1 通り
場が新しいかどうか	2 通り

した。推定精度に影響を与える隠れ層のユニット数や層数は、様々なユニット数・層数を扱った上で、一番妥当であるユニット数 200 と層数 3 に設定した。LSTM の学習方法として、全体からランダムに複数個取得し学習していくミニバッチ学習を用いた。

5.4 出力層の構成

本研究は、対象プレイヤーの手札を推定する事を目的としている。そのため、LSTM の出力層で出力するデータは対象プレイヤーの手札である必要がある。そこで、本研究の LSTM の出力層は各ランクの単体役・複数役・階段役で構成される手役構成 51 項目の手役構成を用意した。単体役はランク 3 から 2 と Joker の 14 項目、複数役はランク 3 の 2 枚・3 枚以上、ランク 4 の 2 枚・3 枚以上…ランク 2 の 2 枚・3 枚以上の 26 項目、階段役はランク 345、ランク 345…ランク K12 の組み合わせからなる 11 項目の構成である。

5.4.1 学習時の所持手札から手役構成への変換

推定対象プレイヤーの所持手札集合から学習時に使用する手役構成には、以下の様に変換する。

- (1) 所持手札集合の中からスート毎に階段役をランクが小さい順 (345,456,567…) から探し、作成出来る場合は所持手札集合から取り出す。
- (2) 所持手札集合の中からランク毎に複数役をランクが小さい順から探し、作成できる場合は所持手札集合から取り出す。
- (3) 所持手札集合の中からランク毎に単体役をランク (3,4…2,Joker) が小さい順から探し、作成できる場合は所持手札集合から取り出す。

5.4.2 推定時の手役構成から所持手札への変換

推定時に出力層から出力された手役構成を用いて、推定対象プレイヤーの手札を以下の様に生成する。

- (1) 51 項目の出力された確率値の中で、一番確率が大きい項目を探す。

- (2) 一番確率が大きい項目のカード組み合わせが未使用カード集合の中に存在していれば、未使用カード集合から取り出す。存在していなければ何もしない。
- (3) 選んだ項目の確率を-100%にする。
- (4) 推定対象プレイヤーの手札枚数に達した場合は終了し、達していない場合は 1 から繰り返す。

5.5 学習方法

LSTM の学習は、1 ゲームを用いて以下の流れに沿って行われる。

- (I) 推定対象プレイヤーのターン時、場の状態と推定対象がカードを提出 or パスした状態を切り取る
- (II) LSTM の入力層に切り取った状態データを入力
- (III) LSTM に入力した際に出力された手役構成と正解となる実際の手役構成の 2 つを用いて、最小二乗法により誤差を出す
- (IV) 生じた誤差を用いて、勾配法 Adam により重みを調整

6. 評価実験

以下の目的で 2 つの実験を行った。

実験 1 未知のデータに対して、提案手法が先行研究よりも手札推定において有効な手法であるか検証する

実験 2 時系列履歴を利用する事が手札推定において有効か検証する事を目的とする

本研究の提案手法の比較対象としては、手札推定の先行研究である柳澤らの手法 [11] を利用している。

6.1 実験 1

6.1.1 目的

未知のデータに対して、提案手法が先行研究の手法よりも有効な手法であるか検証する事を目的とする。

6.1.2 方法

提案手法による手札推定法と先行研究による手札推定法を用いて、用意したゲーム数分手札推定を行わせ、手札推定の精度がどのくらい変化するか検証を行った。提案手法を用いた推定法と先行研究による手札推定法を用いて、1 ゲームが終わるまで以下を繰り返した。

- (I) 推定対象プレイヤーのターン以外の場合、場の状態の更新のみ行う。
- (II) 推定対象プレイヤーのターンの場合、推定対象がカードを提出 or パスした状態を切り取る。
- (III) LSTM の入力層に切り取った状態データを入力した時に出力される手札と先行研究の推定法で出力された手札を保存する。
- (IV) 各推定法で出力した手札を実際に所有していた手札と照らし合わせて、各推定法での推定率を出す。提案手法の学習に用いるトレーニングデータとして、表 2 に載せてある 5 つのクライアントによる 10000 × 10 回対

戦を行って作成した crow の棋譜 100000 ゲーム, paoonR2 の棋譜 100000 ゲーム, wisteria の棋譜 100000 ゲームの三種類を用いた. 手札推定精度の検証を行う際に使用するテストデータとして, トレーニングデータとは別に表 2 に載せてある 5 つのクライアントによる 1000 回対戦を行って作成した crow の棋譜 1000 ゲーム, paoonR2 の棋譜 1000 ゲーム, wisteria の棋譜 1000 ゲームの三種類を用いた. LSTM の学習回数は 50000 回とし, バッチサイズを 40 に設定した. 先行研究の推定法で使用する分類と分類に属するランクごとの出現確率はトレーニングデータと同じデータを用いて作成した.

表 2 使用クライアント一覧

優勝年度	参加階級	クライアント名
2011	無差別級	crow
2012	無差別級	paoonR2
2015	無差別級	wisteria
2015	ライト級	kou2
2016	ライト級	res.kou2

6.1.3 結果

実験結果を以下の表 3, 表 4, 表 5 に示す. 使用枚数は切り取った盤面時に使用されたカード枚数, 盤面数は該当した盤面の数, 提案手法と先行研究は推定率の平均値, F 値と P 値は提案手法と先行研究の手法による一元配置分散分析の結果を示している. 使用枚数 All は全ての盤面データを対象にしている事を示している. また, 各クライアントの使用枚数 All 時の提案手法と先行研究の手法の箱ひげ図を図 2, 図 3, 図 4 に示す. LSTM は提案手法, Pre は先行研究を示している.

表 3 Crow における未知のデータに対する手札推定率

使用枚数	盤面数	提案手法	先行研究	F 値	P 値
All	10863	34.39	23.93	1127.43	5.12E-241
1-18	3752	25.04	19.75	270.43	1.00E-59
19-36	4547	33.40	22.74	647.97	3.91E-138
37-53	2565	49.82	32.15	396.18	5.62E-85

表 4 PaoonR2 における未知のデータに対する手札推定率

使用枚数	盤面数	提案手法	先行研究	F 値	P 値
All	10389	31.75	21.64	1055.49	6.83E-226
1-18	3858	23.67	18.90	216.39	2.49E-48
19-36	4268	30.43	20.62	498.50	2.29E-107
37-53	2264	48.01	28.23	457.03	1.11E-96

表 3~表 5 と図 2~図 4 の結果から, 未知のデータに対する提案手法の推定率は先行研究の推定率よりも高い事が確認出来た. また, 有意水準 0.01 において提案手法の推定率と先行研究の推定率に有意差がある事が確認出来た.

表 5 Wisteria における未知のデータに対する手札推定率

使用枚数	盤面数	提案手法	先行研究	F 値	P 値
All	9659	30.20	19.90	1069.78	2.01E-228
1-18	3905	23.90	18.42	279.10	1.37E-61
19-36	4003	29.25	18.28	558.30	2.24E-119
37-53	1752	46.42	26.88	339.96	1.59E-72

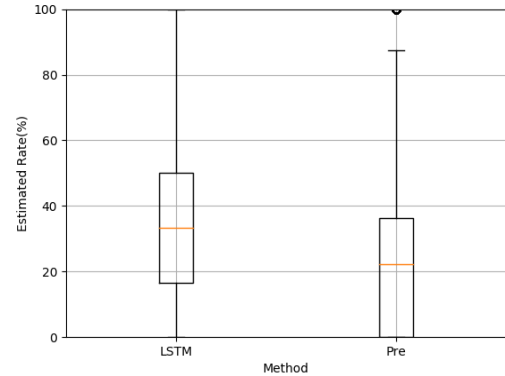


図 2 Crow における使用枚数 All 時の箱ひげ図

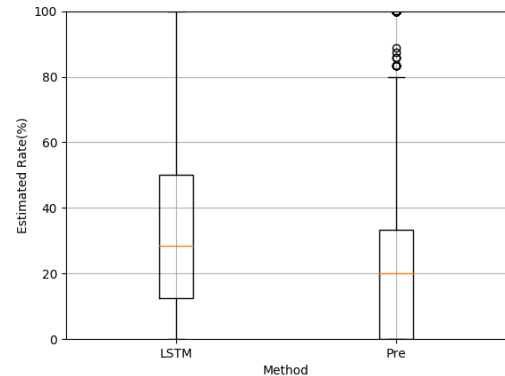


図 3 PaoonR2 における使用枚数 All 時の箱ひげ図

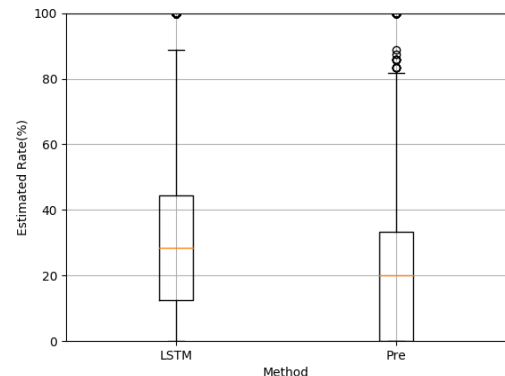


図 4 Wisteria における使用枚数 All 時の箱ひげ図

6.1.4 考察

表 3, 表 4, 表 5 の結果から, 未知のデータに対して, 提案手法は先行研究よりも高い精度で推定が行える事が確認

出来た。このような結果となった理由としては、5章で示した入力層の評価項目がクライアント毎の手札の残し方を学習出来ていたのではないかと考えられる。また、使用カード枚数が増えていくにつれて、提案手法の推定平均率が大きく上昇している事が確認出来た。使用カード枚数が徐々に増えていくという事は、経過ターンが徐々に増えていくことであり、LSTMが利用出来る過去のターン情報も増えていく事でもある。そのため、推定率上昇の理由として、過去のターン情報を用いて推定を行うといったLSTMの効果が発揮されたのではないかと考えられる。

しかし、提案手法と先行研究の手法を比べるとNNの有無が推定精度の違いに関係しているのではないかと考えられる。先行研究の手法は行動履歴を用いてランク毎の出現確率によって手札を推定する手法である。一方、提案手法はLSTMを使用し、高い精度で推定を行う事が出来る地曳らによる相手プレイヤーの提出手役関数[16]はNNを使用している。つまり、提案手法が先行研究よりも高い精度で推定が行えた理由として、時系列履歴を利用した提案手法が有効ではなく、時系列履歴に関係なく単純にNNを用いる事が高い精度で推定を行えた理由として考えられる。

そこで、時系列履歴を利用するLSTMが有効であるか検証するべく、時系列履歴を利用しないNNを用いた比較実験を実験3にて行った。

6.2 実験2

6.2.1 目的

時系列履歴を利用する事が手札推定において有効か検証する事を目的とする

6.2.2 方法

提案手法による推定法(以下提案手法(LSTM)とする)、提案手法のLSTMをNNに切り替えた推定法(以下提案手法(NN)とする)と先行研究による手札推定法を用いて、用意したゲーム数分手札推定を行わせ、手札推定の精度がどのくらい変化するか検証を行った。提案手法(LSTM・NN)を用いた推定法と先行研究による手札推定法を用いて、実験1の(I)(II)(III)と同様の操作を行い、各推定法での推定率を出した。1ゲームが終わるまで以下を繰り返した。LSTM・NNの学習に用いるトレーニングデータとして、表2に載せてある5つのクライアントによる10000×10回対戦を行って作成したcrowの棋譜100000ゲーム、paonR2の棋譜100000ゲーム、wisteriaの棋譜100000ゲームの三種類を用いた。手札推定精度の検証を行う際に使用するテストデータとして、トレーニングデータとは別に表2に載せてある5つのクライアントによる1000回対戦を行って作成したcrowの棋譜1000ゲーム、paonR2の棋譜1000ゲーム、wisteriaの棋譜1000ゲームの三種類を用いた。LSTM・NNの学習回数は50000回とし、バッチサイズを40に設定した。

6.2.3 NNの構成

実験2で用いるNNの入力層のユニット数は表1の評価項目個数である205個であり、出力層のユニット数は各ランクの単体役・複数役・階段役に合わせて51項目用意した。推定精度に影響を与える隠れ層のユニット数や層数は、LSTMと同様のユニット数200と層数3に設定した。NNの学習方法として、全体からランダムに複数個取得し学習していくミニバッチ学習を用いて、5.5と同様の学習を行った。

6.2.4 結果

実験結果を以下の表6、表7、表8に示す。使用枚数は切り取った盤面時に使用されたカード枚数、盤面数は該当した盤面の数、LSTMとNNは推定率の平均値、F値とP値は提案手法(LSTM)との提案手法(NN)による一元配置分散分析の結果を示している。使用枚数Allは全ての盤面データを対象にしている事を示している。また、各クライアントの残り枚数All時の提案手法(LSTM)と提案手法(NN)の箱ひげ図を図5、図6、図7に示す。LSTMは提案手法(LSTM)、NNは提案手法(NN)を示している。そして、提案手法(LSTM)、先行研究と提案手法(NN)の各手法における使用カード枚数ごとの実験結果を図8、図9、図10に示す。縦軸はその手法によって出力された手札と実際の手札の一致率、横軸は推定時に使用されたカード枚数、折れ線グラフは各未使用カード毎の推定率の平均値を表している。

表6 CrowにおけるLSTMとNNの手札推定率

使用枚数	盤面数	LSTM	NN	F値	P値
All	10863	34.39	24.08	1092.78	7.48E-234
1-18	3752	25.04	19.28	308.31	1.13E-67
19-36	4547	33.40	23.13	595.33	2.04E-127
37-53	2565	49.82	32.78	376.24	6.14E-81

表7 PaoonR2におけるLSTMとNNの手札推定率

使用枚数	盤面数	LSTM	NN	F値	P値
All	10389	31.75	24.04	550.22	4.09E-120
1-18	3858	23.67	19.42	153.62	6.02E-35
19-36	4268	30.43	22.92	265.65	7.76E-59
37-53	2264	48.01	33.99	205.50	1.29E-45

表8 WisteriaにおけるLSTMとNNの手札推定率

使用枚数	盤面数	LSTM	NN	F値	P値
All	9659	30.20	22.56	556.10	3.05E-121
1-18	3905	23.90	18.93	222.53	1.22E-49
19-36	4003	29.25	21.47	270.77	7.35E-60
37-53	1752	46.42	33.13	148.08	2.13E-33

表6～表8、図5～図7と図8～図10の結果から、未知のデータに対する提案手法(LSTM)の推定率は提案手法(NN)の推定率よりも高い事が確認出来た。また、有意水準

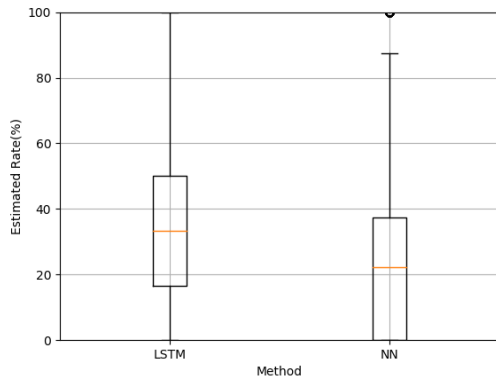


図 5 Crow における使用枚数 All 時の箱ひげ図

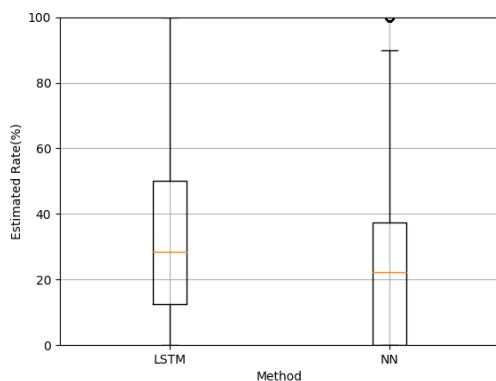


図 6 PaoonR2 における使用枚数 All 時の箱ひげ図

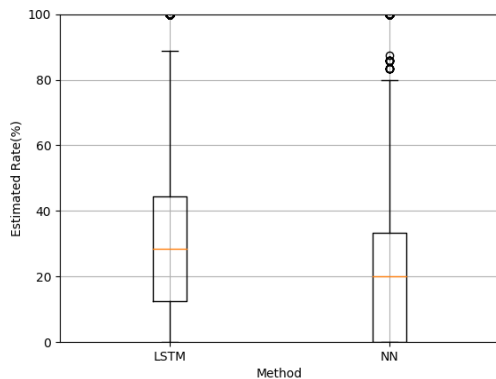


図 7 Wsiteria における使用枚数 All 時の箱ひげ図

0.01 において提案手法 (LSTM) の推定率と提案手法 (NN) の推定率に有意差がある事が確認出来た。

6.2.5 考察

実験結果から、NN を用いた提案手法よりも LSTM を用いた提案手法の方が高い精度で推定が行える事が確認出来た。また、NN に比べて LSTM の方が推定率が大きく上昇している事が分かる。このような結果になった理由としては、手札推定に用いた盤面データは個々に独立したデータではなく、音声データのように時系列の関係があったので

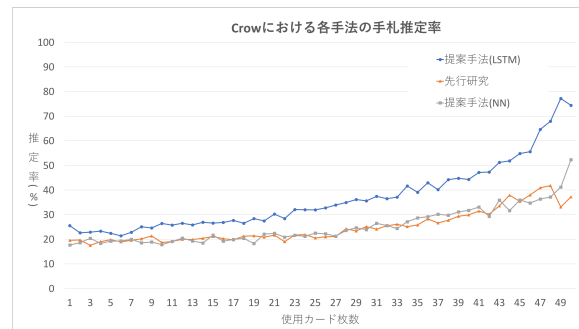


図 8 Crow における各手法の手札推定率

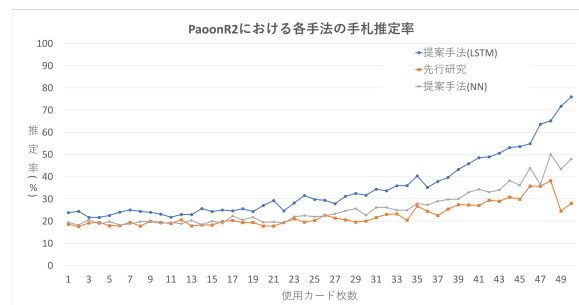


図 9 PaoonR2 における各手法の手札推定率

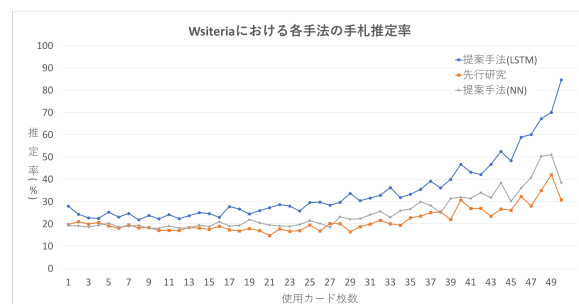


図 10 Wsiteria における各手法の手札推定率

はないかと考えられる。学習データが個々に独立している場合、NN による学習は可能である。しかし、音声データのように学習データに時系列の関係がある場合、NN による学習は困難であるが、LSTM による学習は NN による学習に比べて行いやすい。この事は、Babinec らの研究でも示されている [19]。Babinec らは多層パーセプトロンと RNN の二種類を用いて時系列データ予測を行い、多層パーセプトロンに比べて RNN の方が予測精度が高い事を示した。そのため、NN による手札推定精度が悪く、LSTM による手札推定精度が良い事から、提案手法で用いた盤面データには時系列の関係があったのではないかと考えられる。

以上のことから、単純に NN を利用する事で推定精度が向上することはなく、時系列履歴を用いることが出来る LSTM を利用する事が推定精度の向上に繋がると考えられる。

7. 結論

本研究では、コンピュータ大貧民における場の状況とそ

の時のプレイヤーの行動の2つの時系列履歴を用いた LSTM による相手手札推定法を提案した。提案手法と先行研究による実験の結果、提案手法は先行研究の推定法よりも推定平均率が高く、提案手法の推定率と先行研究の推定率に有意差がある事が確認出来た。この事から、提案手法は先行研究の手札推定法よりも有効な推定法であると示された。また、LSTM と NN による実験の結果、LSTM を用いた提案手法は NN を用いた提案手法よりも推定平均率が高く、LSTM を用いた提案手法の推定率と NN を用いた提案手法の推定率に有意差がある事が確認出来た。この事から、時系列履歴を考慮した LSTM を用いる事で推定精度が向上する事が示された。

今後の課題として、まず、提案手法による手札推定をコンピュータ大貧民 AI に導入した際の勝率の変化を検証する事が挙げられる。また、提案手法の手札推定の汎用性を向上させる点が挙げられる。実力が近いクライアントの棋譜を混合させた棋譜を用いて学習させる事で、提案手法の汎用性が向上するのではないかと考えられる。

謝辞 研究方針や論文執筆にあたり、多くのご指導を賜りました、伊藤毅志研究室の皆様方には、この場を借りて深くお礼申し上げます。

参考文献

- [1] 小谷善行, "第3回将棋電王戦を振り返って:3. コンピュータ将棋の棋力の客観的分析", 情報処理 55(8), Vol2014, pp.851-852, (2014).
- [2] 瀧澤武信, "進化し続けるコンピュータ将棋:1. 最近のコンピュータ将棋の動向", 情報処理 59(2), Vol2018, pp.144-152, (2018)
- [3] Silver, David, et al., "Mastering the game of Go with deep neural networks and tree search.", Nature 529.7587, pp.484-489, (2016).
- [4] 松原仁, "AlphaGo の置き土産", 情報処理 58(8), vol2017, pp.668-669, (2017).
- [5] 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", ゲームプログラミングワークショップ2016 論文集, Vol2016, pp.188-195, (2016).
- [6] 築地毅, 柴原一友, "CNN 麻雀 - 麻雀向け CNN 構成の有効性-", ゲームプログラミングワークショップ2017 論文集, Vol2017, pp163-170, (2017).
- [7] 近藤まなみ, 松本啓之亮, 森直樹, "深層学習を用いた人狼エージェントの行動予測", システム制御情報学会研究発表講演会講演論文集, 61 巻, 4p, (2017).
- [8] 大久保誠也, 小林正人, 大久保誠也, 本多武尊, 眞鍋秀聡, 青木輝人, 柿下容弓, 小松原頌之, 西野 哲朗, "第1回コンピュータ大貧民大会 (UECda-2006) の報告", 情報処理学会ゲーム情報学研究報告, Vol.GI-17, pp.25-32, (2007).
- [9] 須藤郁弥, 篠原歩, "モンテカルロ法を用いたコンピュータ大貧民の試行ルーチン設計", 第2回 UEC コンピュータ大貧民シンポジウム講演予稿集, 電気通信大学, (2009).
- [10] 吉原大夢, 大久保誠也, "コンピュータ大貧民における手札推定の有効性について", 情報処理学会研究報告, Vol.2013, No.4, pp.1-6, (2013).
- [11] 柳澤佑介, 松崎公紀, "大貧民における出現頻度と提出手

- 役履歴を用いた相手手札推定", 情報処理学会研究報告, Vol.2015, No.9, pp.1-6, (2015).
- [12] 須藤郁弥, 成澤和志, "UEC コンピュータ大貧民大会向けクライアント「snow」の開発", 第2回 UEC コンピュータ大貧民シンポジウム講演予稿集, 電気通信大学, (2010).
 - [13] 小沼 啓, 本多 武尊, 保木 邦仁, 西野 哲朗, "コンピュータ大貧民に対する差分学習法の応用", 研究報告ゲーム情報学 (GI), Vol2012, pp1-4, (2012).
 - [14] "UEC 標準ルール", UECda2017, 入手先 (http://www.tnlab.inf.uec.ac.jp/daihinmin/2017/document_rules.html)(2018-01-29).
 - [15] Gers, Felix A., Schmidhuber, J., Jürgen, and Cummins, Fred, "Learning to forget: Continual prediction with LSTM", Neural computation, Vol.12, No.10, pp.2451-1471, (2000).
 - [16] 地曳隆将, 松崎公紀, "大貧民における棋譜データからの提出手役評価関数の学習", 情報処理学会研究報告, Vol.2014, No.15, pp.13-34, (2014).
 - [17] Yoshua Bengio, "Practical Recommendations for Gradient-Based Training of Deep Architectures", Neural Networks: Tricks of the Trade, pp.437-478, (2012)
 - [18] Y Bengio, P Simard and P Frasconi, "Learning long-term dependencies with gradient descent is difficult", IEEE transactions on neural networks, pp.157-166, (1994)
 - [19] S.Babinec, and J.Pospchal, "Merging echo state and feedforward neural networks for time series forecasting" Artificial Neural Networks ICANN 2006, pp.367-375, (2006)