SVRを適用したFALCONによる トランプゲームに対する学習実験†

笠原 和真*¹・二本 健太*²・伊藤 崇*¹・高橋 健一*¹・稲葉 通将*¹

本論文では、多人数不完全情報ゲームであるトランプゲームのハーツにおいて、FALCON（a Fusion Architecture for Learning, COgnition, and Navigation）を用いた学習エージェントの学習精度を向上させるための手法を提案している。FALCONは、強化学習の手法の1つであるファジィ ART(Adaptive Resonance Theory)を拡張した機械学習手法である。知覚、行動、報酬の全てのベクトルに対して同時に複数のマッピングを学習することにより、知覚状態空間の離散化及び行動規則の学習を同時に行うことができる。既存研究において、トランプゲームのハーツに対するFALCONの有効性が示され、あらかじめ設定されたルールに基づいて行動するエージェントに勝つことに成功している。しかしながら、モンテカルロシミュレーションにより行動を決定するエージェントに勝つことはできなかった。本研究では、学習エージェントの性能向上のための改良として、実際のハーツにおける戦略に基づいた行動種類の変更と、学習時における行動選択に対してサポートベクター回帰による予測を適用する手法を提案する。これらの改良を用いる場合と用いない場合とで学習実験を行い、学習エージェントの性能の比較を行う。

キーワード：強化学習、ファジィ ART、FALCON、多人数不完全情報ゲーム、サポートベクター回帰

1. はじめに

近年、人工知能を用いた研究が多くなされ、その技術はゲームやロボットなどの分野で広く用いられている [1]。最近では、Google DeepMind 社が開発した囲碁の人工知能『AlphaGo』の進化版である『Master』が、世界トップレベルの棋士を相手に無敗の戦績を上げ大きな注目を集めた [2]。また、人間の対局データを用いず自己対戦のみで学習を行う『AlphaGo Zero』が発表されている [3]。このように、人工知能の技術をゲームに応用した研究の1つとして、囲碁やチェスといった二人完全情報ゲームが注目されている。二人完全情報ゲームとは、プレイヤーが2人で、局面情報が全てのプレイヤーに与えられているゲームである。これに対して、3人以上のプレイヤーが存在し、局面に不完全な情報があるゲームを多人数不完全情報ゲームという。一般的に、不完全情報ゲームは完全情報ゲームよりも考慮すべき要素が多く、また確率的な要素も多分に含まれている。したがって、多人数不完全情報ゲームは不確定な情報を多く含む実世界に近い問題と言える。このような環境に対して効率的に強化学習を適用するためには、知覚状態空間を適切に離散化する必要がある。状態空間を離散化する手法は、タイルコーディング、ニューラルネットワーク、ファジィ推論などのオフライン手法、及び ART (Adaptive Resonance Theory) などのオンライン手法に大別することができる [4]。

本研究では、Ah-Hwee Tan によって提案されている FALCON（a Fusion Architecture for Learning, COgnition, and Navigation）[5, 6] を用いて多人数不完全情報ゲームであるカードゲームのハーツを解く。FALCON は ART を拡張した機械学習手法であり、二人完全情報ゲーム及び多人数不完全情報ゲームに対して有効性が示されている [7, 8]。関連研究として、二人完全情報ゲームで有効性が示されている UCT（Upper Confidence bound applied to Trees）アルゴリズムをマルチプレイヤーゲームに拡張し分析を行った研究 [9] や、線形関数近似により事前知識を用いない不完全情報ゲームのための Q 学習が提案されている [10]。前者の研究では、UCT アルゴリズムを用いてマルチプレイヤーゲームを解くために、実際のゲームでは公開されない相手の手札などの情報を公開情報とし、完全情報ゲームとして実験を行っている。UCT アルゴリズムなどの予測を用いた手法によって不完全情報ゲームを解く場合、予測の精度を向上させるために様々な可能性を考慮する必要があり、計算量がとても膨大になる。また、後者の研究で用いている Q 学習においても、情報量の多い不完全情報ゲームを解くためには状態空間が膨大になり学習に時間がかかる。本研究で用いる FALCON はオンライン学習手法であり、知覚、行動、報酬の全てのベクトルに対して同時に複数のマッピングを学習することにより、知覚情報空間の離散化及び行動規則の学習を同時に実施することができる。ハーツに対する FALCON の有効性については森下によって示されている [6]。そして、その改良手法を提案し、より強い学習エージェントを構築することに成功している [11]。また、FALCON を用いて訓練した学習器を複数個組み合わせることで、より強いエージェントを構築する手法も提案されている [12]。しかし、モンテカルロシミュレーションにより行動を選択するエージェントに勝つことはできなかった。

そこで本研究では、FALCON の改良手法を提案し、より強

† Learning Experiments of a FALCON Using SVR for a Card Game
Kazuma KASAHARA, Kenta NIMOTO, Takashi ITO,
Kenichi TAKAHASHI, and Michimasa INABA

*1 広島市立大学大学院
Graduate School of Information Sciences, Hiroshima City University

*2 株式会社日立インフォメーションエンジニアリング
Hitachi Information Engineering

学習エージェントを構築することを目的とする。まず、エージェントをより戦略的に行動させるため、FALCON の行動種類を実際のハーツにおける戦略に基づいて再設定した。次に、効率的な学習を行うため、FALCON の学習時における行動選択に対して回帰分析手法を用いた予測を適用した。回帰分析手法には、行動に対する評価を危険率で求めるためにサポートベクター回帰を採用した。サポートベクター回帰は多くの研究に用いられており、オープンソースの機械学習ライブラリ『LIBSVM』[13]によって容易に使用することができる。提案手法を用いた場合と用いない場合とでそれぞれ学習実験を行い、学習エージェントの性能がどのように変化するか確認した。

本論文は以下のような構成になっている。2章では本研究で用いたハーツのルールについて述べる。3章では FALCON を用いた強化学習について、4章では予測のために FALCON に導入したサポートベクター回帰 (SVR) について説明する。5章では、本論文の提案する行動種類の変更と、SVR を適用した FALCON について説明している。6章では、提案した手法と従来の手法との比較実験を行い、考察を述べる。最後に、7章でまとめを述べる。

2. ハーツ

2.1 ルール

トランプゲームのハーツは一般的に 4 人でプレイされ、ジョーカーを除いた 52 枚のカードを用いる。スートごとに A, K, Q, ..., 4, 3, 2 の順で強さの順位付けがされており、スート間の優劣はない。各プレイヤーはゲームのはじめに 13 枚のカードを手札として持ち、自分の番には必ず場に 1 枚カードを出す。親プレイヤーから始めて全員が 1 枚ずつカードを出すことをトリックと呼び、13 トリック連続して行うことで 1 ゲームが終了する。また、各トリックで最初に出されるカードのことをリーディングカードと呼び、そのときカードを出すプレイヤーを親と呼ぶ。親以外のプレイヤーを子と呼ぶ。子は、リーディングカードと同じスートを場に出す。同じスートのカードがない場合に限り、任意のカードを出すことができる。全てのプレイヤーがカードを場に出し終えた時点で、リーディングカードと同じスートの中の最も強いカードを出したプレイヤーが、場のカードを全て引き取り、次のトリックの親となる。1 ゲーム終了後、各プレイヤーの引き取ったカードのうち、ハートは 1 枚 1 点、スペードの Q は 1 枚 13 点として各プレイヤーの罰点を計算する。罰点を少なくすることが各プレイヤーの目的である。本研究では、一般的に用いられる「最初のトリックの前に行う手札交換」と「シュート・ザ・ムーンを含む得点のバリエーション」の 2 つのルールを無視した。

2.2 対戦に用いるエージェント

本研究では、学習エージェントの学習及び評価実験における対戦相手として、ルールベースエージェント、モンテカルロエージェントの 2 種類のエージェントを用いた。ルールベースエージェントは gnome-hearts[14] を元にしたルールを搭載しているエージェントである。自分の所持しているカードの種

図 1 FALCON の構成

類や場の状態、未提出のカードの情報などを考慮して、設定された if - then ルールに基づき出し札を決定する。モンテカルロエージェントは UCT モンテカルロ法 [15] により FALCON に設定した行動種類から行動を選択するエージェントである。行動の探索は、現在のトリックをシミュレート対象とし、残っているカードを 3 人の相手プレイヤーにランダムに配り、UCT モンテカルロ法に基づき行動選択を行い、プレイする。その後、相手プレイヤーの行動をランダムに行い、トリック終了時の獲得罰点を記録する。このシミュレーションを 200 回行い、最も獲得罰点が少なかった行動種類を選択する。事前に行った実験により、これら 2 種類のエージェントの中ではモンテカルロエージェントが最も強く、次にルールベースエージェントが強いということが分かっている [11, 12]。

3. FALCON による強化学習

3.1 FALCON の構成

図 1 に、FALCON の構成図を示す。sensory field (SF), motor field (MF), feedback field (FF) はそれぞれ cognitive field (CF) と関連付けられている。エージェントが M 個の知覚センサを有するとき、SF には知覚ベクトル $S = (s_1, \dots, s_M)$ が与えられる。MF には行動ベクトル $A = (a_1, \dots, a_K)$ が与えられる。 K は選択可能な行動の数を表し、 a_i ($i = 1, \dots, K$) の値は i 番目の行動が選択されたときに 1、それ以外の場合は 0 となる。FF には報酬ベクトル $R = (r, 1-r)$ が与えられる。 r はエージェントが環境から受け取る報酬であり、 $r \in [0, 1]$ である。CF は L 個のニューロン n_j ($j = 1, \dots, L$) を有し、SF, MF 及び FF のニューロンとそれぞれ重みベクトル $W^s_j = (w^s_{1j}, \dots, w^s_{Mj})$, $W^m_j = (w^m_{1j}, \dots, w^m_{Kj})$ 及び $W^f_j = (w^f_{1j}, w^f_{2j})$ によって関連付けられている。重みベクトルの要素 $w^y_{xj} \in [0, 1]$ は、ニューロン n_j と SF, MF 及び FF のニューロンとの関連の強さを表す。FALCON は重みベクトルを更新することによって、知覚、行動、報酬の関連を学習する。

3.2 行動の選択と学習

行動選択フェーズにおいて、行動と重みベクトルの選択強度が式 (1) により計算される。

$$T_j = \gamma_s \frac{\|S \wedge W^s_j\|}{\alpha_s + \|W^s_j\|} + \gamma_m \frac{\|A \wedge W^m_j\|}{\alpha_m + \|W^m_j\|} + \gamma_f \frac{\|R \wedge W^f_j\|}{\alpha_f + \|W^f_j\|} \quad (1)$$

二項演算子 $\wedge$ はファジィ AND オペレータであり、 $(X \wedge W_j) = (\min(x_1, w_{1j}), \min(x_2, w_{2j}), \dots, \min(x_n, w_{nj}))$ で定義される。ノルム $\|\cdot\|$ は各要素の絶対値和である。パラメータ $\gamma_s, \gamma_m, \gamma_f$ は非負の実数であり、 $\gamma_s + \gamma_m + \gamma_f = 1$ を満たす。パラメータ $\alpha_s, \alpha_m, \alpha_f$ は非負の実数である。また、式 (1) により選択強度が計算される際、行動ベクトル A は全て 1、報酬ベクトル $R = (1, 0)$ とした。さらに、予備実験によりパラメータ調整を行い、 $\gamma_s = 1.0, \gamma_m = 0.0, \gamma_f = 0.0$ とした。式 (1) から選択強度が最大となるニューロンが n_j として選択され、重みベクトル W^m_j から $w^m_{kj} (k = 1, \dots, K)$ の値が最大となるような行動 k が、エージェントが実行する行動として選択される。

学習フェーズにおいては、環境からのフィードバックが正であれば、システムは知覚ベクトル S 、行動ベクトル A 、報酬ベクトル R の関連を強くする。負であれば関連を弱くする。 S と W^s_j 、 A と W^m_j 、 R と W^f_j の類似性がそれぞれ式 (2)、式 (3)、式 (4) によって計算される。

$$\frac{\|S \wedge W^s_j\|}{\|W^s_j\|} \geq \rho_s \quad (2)$$

$$\frac{\|A \wedge W^m_j\|}{\|W^m_j\|} \geq \rho_m \quad (3)$$

$$\frac{\|R \wedge W^f_j\|}{\|W^f_j\|} \geq \rho_f \quad (4)$$

式 (2)、式 (3)、式 (4) が全て満たされている場合、式 (5)、式 (6)、式 (7) によって重みベクトルは更新される。 ρ_s, ρ_m, ρ_f は閾値であり、 $\beta_s, \beta_m, \beta_f$ は学習速度であり $0 \leq \beta_s, \beta_m, \beta_f \leq 1$ を満たす。通常 $\beta_s, \beta_m, \beta_f$ の値は 1 であり、このとき高速学習であるという。値が 1 未満の場合、重みベクトルは現在の値を考慮しながら徐々に更新される。

$$W^s_j = (1 - \beta_s)W^s_j + \beta_s(S \wedge W^s_j) \quad (5)$$

$$W^m_j = (1 - \beta_m)W^m_j + \beta_m(A \wedge W^m_j) \quad (6)$$

$$W^f_j = (1 - \beta_f)W^f_j + \beta_f(R \wedge W^f_j) \quad (7)$$

式 (2)、式 (3)、式 (4) が 1 つでも満たされていない場合、次に選択強度が最大となるカテゴリを探し、条件を全て満たす場合、重みの更新を行う。条件を満たすようなカテゴリが存在しない場合、 $W^s_{new} = S, W^m_{new} = A, W^f_{new} = R$ として新しいカテゴリを追加する。

3.3 FALCON を用いたハーツのための学習

FALCON を用いて学習を行うためには、行動種類と知覚情報を設定する必要がある。本研究ではハーツによる実験を行うため、行動種類は出し札を決定するための手続き、知覚情報は局面状態とした。設定した知覚情報を表 1 に示す。表 1 中の、par1 から par3 はエージェントが親番時に使用する知覚情報であり、chi1 から chi5 は子番時に使用する知覚情報である。ま

表 1 設定した知覚情報

ID	知覚情報	ビット数
par1	♠Q, K, A の状態 (自分が所持, 相手が所持, プレイ済み).	6
par2	自分が ♠J 以下をもっているか.	1
par3	相手が ♠J 以下をもっているか.	1
chi1	Breaking Hearts が行われたか.	1
chi2	♠Q, K, A の状態 (自分が所持, 相手が所持, プレイ済み).	6
chi3	リーディングカードのスート (以下対象スートとする).	2
chi4	対象スートの所持枚数 (0, 1, 2 枚以上).	3
chi5	対象スートに関して、手札の最強カードが残っているもので最も強いカードであるか.	1

た、各知覚情報は、0 または 1 の値を取る。よって、親番時には計 8 次元、子番時には計 13 次元の状態空間を用いている。学習における重みの更新方法は既存研究 [12] と同様なものを用いた。学習において、FALCON を用いた学習エージェントは、選択可能な行動種類の中から重みの値が最大である行動を選択して実行する。そして環境からフィードバックを受け取り、行動ベクトルの値を更新することで学習が進行する。

フィードバックについても、既存研究 [11, 12] と同様に、獲得罰点を用いた。また、トリック別、かつ手番別にそれぞれ独立した 52 個の FALCON を用いて学習を行う手法 [11] を適用した。

4. サポートベクター回帰

2 値分類問題を解くために Vapnik らによって提案された教師あり学習によるパターン認識モデルの 1 つとして、サポートベクターマシン (Support Vector Machine : SVM) がある [16]。また、SVM を回帰に適用し、回帰関数を学習する手法をサポートベクター回帰 (Support Vector Regression : SVR) という。

SVR では、 N 個の特徴ベクトル x_i に対して、制約条件式 (8) の下、式 (9) を最小化するパラメータを求める問題である。ここで、 w は最適超平面を表す係数ベクトル、 ϵ はインセンシティブ帯、 ξ は各学習データに対する誤差であり、パラメータ C は、第 1 項のマージンの大きさと第 2 項の学習データの損失とのバランスを決める定数である。

$$\begin{aligned} ((w^T x_i) + b) - y_i &\leq \epsilon + \xi_i, (i = 1, \dots, N) \\ y_i - ((w^T x_i) + b) &\leq \epsilon + \xi_i^*, (i = 1, \dots, N) \\ \xi_i, \xi_i^* &\geq 0, (i = 1, \dots, N) \end{aligned} \quad (8)$$

$$\frac{1}{2} \|w\|^2 + \frac{C}{N} \sum_{i=1}^N (\xi_i + \xi_i^*) \quad (9)$$

本研究では、SVR を適用するにあたりオープンソースの機械学習ライブラリの 1 つである『LIBSVM』(Version 3.22) [13] を利用した。SVM 及び SVR に関する複数の手法が提供されているが、実験では ν -SVR という手法を利用した [17]。

5. 提案手法

5.1 行動種類の変更

既存研究 [6, 11, 12] では、罰点札である ♠Q と ♡のカードや、手札内でのカードの強弱関係を考慮した行動種類を設定していた。こうすることで、ハーツにおける基本的な行動選択規則を獲得することができ、ルールベースエージェントを打ち負かすことに成功した。しかしながら、モンテカルロシミュレーションにより行動を決定するモンテカルロエージェントには勝つことができなかった。その原因は、人間のような戦略的な行動規則、特に駆け引きを伴うような行動規則を獲得することができなかったからと考えられる。そこで、実世界のハーツにおける戦略について調査し、学習エージェントの行動種類に取り入れた。本研究では、ハーツに関して詳細にまとめられた『Mark's Hearts Tips』[18] 及び、ハーツのプレイについて詳細にまとめられた『ハーツ攻略』[19] を参考にして次の 5 つのテクニックを採用した。

1. Void：あるスートのカードが手札に 1 枚も無い状態になるようカードを出すプレイ。
2. Smoke Out：同じスートを何度もリードし、特定のカードを出すように誘導するプレイ。
3. Cash：強いカードを出し意図的にトリックを取るプレイ。
4. Exit：弱いカードを出しトリックを取らないようにするプレイ。
5. Spear Play：♠Q でリードするプレイ。

本研究で用いた行動種類を表 2 に示す。表 2 において par1～par18 は親番にのみ選択される。chi1～chi5 は子番のリーディングカードと同じスートのカードがある場合にのみ、chi'1～chi'6 は無い場合にのみそれぞれ選択される。par3～par6 と chi'4, chi'5 は Void を考慮し新しく追加した。par1 は Spear Play を考慮し新しく追加した。以前から存在するが、par2 は Smoke Out に、par15～par18 と chi3 は Cash に、par7～par14 と chi2, chi5 は Exit にそれぞれ対応している。

5.2 SVR を適用した FALCON

5.2.1 SVR を適用した FALCON の構成

既存研究 [11, 12] における FALCON の学習において、ゲーム開始から弱いカードを優先して出し、序盤で弱いカードを出しきってしまうために終盤でゲームが不利になる傾向が多く見られた。これはゲームの性質上、弱いカードを優先的に出すという行動が罰点を取りにくいために正のフィードバックを受け取りやすく、そのような行動が選ばれるように学習が進むためである。そこで本研究では、カードの強弱のみで行動を決めるのではなく、現在の局面情報に対する各出し札の将来的な危険度を SVR を用いて推定し、それを用いて行動を決定する手法を提案する。ここで将来的な危険度とは、現在以降のトリックにおいて ♠Q を獲得するか否かを表す尺度（以下、DegSQ と表す）である。SVR を適用した FALCON の構成を図 2 に示す。SVR を適用した FALCON では、通常の FALCON と同様に、はじめに選択可能な行動種類の中から重みの値が最大であ

表 2 設定した行動

ID	行動
par1	♠Q を出す。
par2	Q より弱い ♠ のうち最強の ♠ を出す。
par3	♡の所持枚数が 2 枚以下の時に、手札の中で最弱の ♡ を出す。
par4 ～ par6	♠, ♡, ♣ の所持枚数が平均よりも少ない時に、相手が所持している最強の ♠, ♡, ♣ よりも弱く、手札の中で最強の ♠, ♡, ♣ を出す。
par7 ～ par10	相手が所持している最強の ♡, ♠, ♡, ♣ よりも弱く、手札の中で最強の ♡, ♠, ♡, ♣ を出す。
par11 ～ par14	手札の中で最弱の ♡, ♠, ♡, ♣ を出す。
par15 ～ par18	手札の中で最強の ♡, ♠, ♡, ♣ を出す。
chi1	♠K, A を 1 枚でも所持し、♠Q が場に無い時に、♠K, A のうち強い方を出す。
chi2	場にあるリーディングカードと同じスートのカードの中で最強のカードよりも弱く、手札の中で最強のカードを出す。
chi3	手札の中で最強のカードを出す。
chi4	手札に 7～J のカードが 1 枚以上ある時に、小さい方から優先して出す。
chi5	手札の中で最弱のカードを出す。
chi'1	♠Q を出す。
chi'2	♠K, A を 1 枚でも所持し、♠Q が残っている場合に、♠K, A のうち強い方を出す。
chi'3	手札の中で最強の ♡ を出す。
chi'4 ～ chi'5	♡, ♣ の所持枚数が 2 枚以下の時に、手札の中で最強の ♡, ♣ を出す。
chi'6	♡, ♠, ♡, ♣ の順に優先して、手札の中で最強のカードを出す。

る行動を選択する。次に選択された行動を SVR によって逐次的に評価し、その行動に対する DegSQ を推定する。具体的には、決定された出し札の情報と FALCON に設定した知覚情報の特徴とし、LIBSVM を利用し SVR により DegSQ を推定する。もし、推定された DegSQ があらかじめ設定した閾値よりも小さければその行動を実行し、そうでなければ、DegSQ が閾値よりも小さくなるまで、次に重みの大きな行動を選択し DegSQ を推定するまでを繰り返す。また、DegSQ が閾値よりも小さくなるような行動が存在しない場合は、FALCON が最初に選択した行動を実行する。

5.2.2 SVR 学習モデルの作成

LIBSVM を用いて学習を行うためには、ラベルと特徴から成る訓練用データセットを用意する必要がある。モンテカルロエージェント 4 体を 100,000 ゲーム対戦させ対戦ログを記録し、それを用いて訓練用データセットを作成した。ラベルは、あるトリックにおいて、そのトリック以降に ♠Q を獲得する場合は 1, しない場合は -1 とした。つまり、DegSQ が 1 に近い

図2 SVR を適用した FALCON の構成

表3 SVR に与える局面情報

No	知覚情報	ビット数
1	各スートについて自分が所持しているか.	4
2	♡を提出することができるか.	1
3	♠Q, K, A の状態 (自分が所持, 相手が所持, プレイ済み).	6
4	自分が ♠J 以下をもっているか.	1
5	相手が ♠J 以下をもっているか.	1
6	各スートについて, 自分の最弱よりも弱いカードの残り枚数.	4
7	各スートについて, 自分の最弱よりも強いカードの残り枚数.	4
8	各スートについて, 自分の最強よりも弱いカードの残り枚数.	4
9	各スートについて, 自分の最強よりも強いカードの残り枚数.	4

ほど将来的に ♠Q を獲得する確率が高く, -1 に近いほど獲得しない確率が高いと言える。また特徴は, 出し札のスートとランクを表す 5 次元と, 表 3 に示す 29 次元の計 34 次元のベクトルを用いた。

データセットは手番及びトリック別のグループに分け生成した。また, 各グループに対して, ラベルごとに 10,000 件のデータセットを収集した。ただし, ゲームの性質上起こりにくい局面が存在するため, データセット数が 10,000 件に満たない場合はデータセット数が少ないラベルに合わせてアンダーサンプリングして用いた。LIBSVM には, データセットをスケーリングするためのプログラムが提供されている。このプログラムを用いて生成したデータセットをスケーリングし, スケーリング後のデータセットを訓練用データセットとして用いた。また, 学習には ν -SVR を使用し, カーネルの種類は式 (10) を用いた RBF とした。

$$K(x, x') = \exp(-\gamma \|x - x'\|^2) \quad (10)$$

表4 採用した ν の値

トリック	親	1 番目の子	2 番目の子	3 番目の子
1	0.40	0.40	0.40	0.80
2	0.10	0.10	0.10	0.60
3	0.15	0.50	0.30	0.60
4	0.55	0.30	1.00	0.60
5	0.25	0.50	0.40	0.50
6	0.50	0.90	0.60	0.20
7	0.10	0.60	0.50	0.20
8	0.80	0.30	0.50	0.40
9	0.50	0.80	0.50	0.40
10	0.50	0.50	0.30	0.50
11	0.55	0.10	0.30	0.40
12	0.50	0.20	0.50	0.50
13	0.50	0.50	0.50	0.50

表5 採用した ν を用いた場合の予測性能 (F 値と精度)

トリック	親	1 番目の子	2 番目の子	3 番目の子
1	0.26 (0.58)	0.36 (0.37)	0.39 (0.51)	0.56 (0.68)
2	0.39 (0.48)	0.43 (0.56)	0.34 (0.54)	0.52 (0.70)
3	0.37 (0.53)	0.32 (0.48)	0.38 (0.54)	0.51 (0.68)
4	0.28 (0.62)	0.33 (0.49)	0.28 (0.61)	0.60 (0.74)
5	0.43 (0.63)	0.36 (0.61)	0.31 (0.61)	0.55 (0.73)
6	0.50 (0.69)	0.27 (0.56)	0.36 (0.60)	0.54 (0.82)
7	0.52 (0.74)	0.41 (0.66)	0.34 (0.69)	0.48 (0.75)
8	0.50 (0.71)	0.37 (0.76)	0.32 (0.64)	0.38 (0.81)
9	0.45 (0.76)	0.39 (0.78)	0.36 (0.72)	0.23 (0.78)
10	0.62 (0.86)	0.30 (0.81)	0.20 (0.81)	0.42 (0.87)
11	0.68 (0.91)	0.45 (0.89)	0.26 (0.84)	0.27 (0.87)
12	0.84 (0.96)	0.15 (0.92)	0.11 (0.87)	0.22 (0.91)
13	1.00 (1.00)	- (0.97)	0.13 (1.00)	0.25 (0.98)

ここで, カーネル関数のパラメータ γ は $1/(\text{特徴数})$, 式 (9) におけるパラメータ C は 1 としている。これらの値は LIBSVM にデフォルトで設定されている値である。 ν -SVR のパラメータ ν は, 次のような実験から決定した。はじめに, 手番およびトリックごとに学習モデルを生成する。つまり, 各トリックと各手番に対応した 52 個の学習モデルが生成される。その後, モンテカルロエージェント 4 体を対戦させ, 各プレイヤーの各トリックにおける DegSQ の予測を記録し, ゲーム終了時に予測が正しかったか否かを確かめる。 ν を 0.05 から 1.0 まで 0.05 刻みに変化させながらこの対戦実験を行い, 10,000 ゲームの結果から ♠Q を獲得する予測に対する F 値 (= $(2 \cdot \text{精度} \cdot \text{再現率}) / (\text{精度} + \text{再現率})$) を算出する。

このような実験を通し, 各トリックと各手番において F 値が最大となる ν を学習実験に用いる値として採用した。各トリック, 各手番に対する採用した ν の値を表 4 に, その場合における予測性能を表 5 にそれぞれ示す。

表 5 では, 括弧外の数値は将来的に ♠Q を獲得する予測に対する F 値を, 括弧内の数値は予測全体に対する精度をそれぞれ表す。- (ハイフン) は, 算出できなかった部分を表す。表 5 より, 親番ではトリックの経過に伴い, F 値も精度も増加傾向にあることがわかる。これは, トリックが経過するほど考慮す

表 6 学習ゲーム数が 100,000 の場合における閾値による学習後の学習エージェントの平均獲得罰点比率

閾値	ルールベース エージェントと 対戦した場合	モンテカルロ エージェントと 対戦した場合
-1.0	0.2215	0.2548
-0.8	0.2187	0.2543
-0.6	0.2177	0.2537
-0.4	0.2193	0.2543
-0.2	0.2179	0.2547
0.0	0.2182	0.2545
0.2	0.2200	0.2570
0.4	0.2184	0.2552
0.6	0.2175	0.2564
0.8	0.2187	0.2531
1.0	0.2185	0.2520

べき要素が少なくなり予測が容易になるためであると考えられる。一方、子番では精度は増加しているが F 値は減少傾向にある。これはゲームの性質上、終盤の子番で ♠Q を獲得することは少なく、♠Q を獲得するという予測が的中しにくくなってしまったためであると考えられる。実際に、精度は高くなっているため全体的には精度の高い予測ができていることが分かる。

6. 実験

5 章で紹介した提案手法の有効性を確かめるために、2.1 節で紹介したルールに基づくハーツの対戦実験を行った。エージェントの評価方法として、ハーツにおける強さ、つまり平均獲得罰点比率の低さを比較する。平均獲得罰点比率とは、4 体のエージェントの合計罰点 (1 ゲームあたり 26 点) に対するエージェントの獲得罰点の比率である。既存研究 [11] のプログラムに対し、本研究における提案手法をすべて適用した場合の学習実験の方法を以下に示す。まず、学習エージェントをルールベースエージェント 3 体と N ゲーム対戦させ学習させる。学習後、学習エージェントをルールベースエージェント 3 体及びモンテカルロエージェント 3 体と 5,000 ゲーム対戦させ平均獲得罰点比率を測定する。ただし、学習後の行動選択において SVR は使用しない。また、SVR のパラメータについては 5.2 節のものをを用いる。これを 10 回繰り返して行い、平均獲得罰点比率を算出する。

6.1 DegSQ の閾値比較実験

5.2 節で述べた DegSQ の閾値を設定するための実験の結果を示す。学習ゲーム数 N を 100,000 とし、閾値を -1.0 から 1.0 まで 0.2 ずつ変化させ、平均獲得罰点比率を比較した。比較を表 6 に示す。表 6 より、閾値として、1.0 を用いた場合のモンテカルロエージェントに対する学習エージェントの平均獲得罰点比率が最も低いという結果であった。したがって、DegSQ に対する閾値は 1.0 とする。

6.2 学習ゲーム数による比較

学習ゲーム数 N の値を 0, 1, 100, 1,000, 10,000, 以降 10,000

図 3 学習後にルールベースエージェント 3 体と対戦させた場合における学習ゲーム数 N と学習エージェントの平均獲得罰点比率の関係

図 4 学習後にモンテカルロエージェント 3 体と対戦させた場合における学習ゲーム数 N と学習エージェントの平均獲得罰点比率の関係

刻みで 100,000 まで変化させ、提案手法を適用する場合としない場合のそれぞれについて実験を行った。図 3 に学習後にルールベースエージェント 3 体と対戦した場合、図 4 に学習後にモンテカルロエージェント 3 体と対戦した場合の学習エージェントの平均獲得罰点比率をそれぞれ示す。

図 3 及び図 4 より、平均獲得罰点比率は提案手法を適用した場合の方が低くなっていることがわかる。ルールベースエージェントと対戦する場合は、既存研究 [11, 12] においても学習エージェントの平均獲得罰点比率は 0.25 を下回りルールベースエージェントに勝つことが可能であった。提案手法を適用することで、更に平均獲得罰点比率を減少させることができた。また、モンテカルロエージェントと対戦する場合は、既存研究 [11, 12] においては学習エージェントの平均獲得罰点比率が 0.25 を下回ることではできなかった。提案手法を適用することで、平均獲得罰点比率を減少させることができ、0.25 を下回り勝利する場合も見られた。

次に、学習ゲーム数 N を 100,000 に設定した際の、学習によ

り生成されたカテゴリ数について表 7 に示す。表 7 より、ゲームの序盤や終盤は、似た局面が多く現れるため、生成カテゴリ数が多い。また、ゲーム中盤では、似た局面が現れることが少なく、生成されるカテゴリは少ないことが分かる。

6.3 手法による比較

本研究では大きく分けて 2 つの手法を提案したが、この 2 つの提案手法がそれぞれのくらい学習エージェントの性能向上に貢献しているかを確認するための実験も行った。既存研究 [11] のプログラムに対し行動種類の変更のみを行った場合と、SVR の適用のみを行った場合それぞれにおいて、上記と同様な実験を行った。そして、学習ゲーム数 N が 100,000 であるときの、学習後の学習エージェントの平均獲得罰点比率を表 8 にまとめる。また、表 8 中の括弧内は、10 試行中に学習エージェントの平均獲得罰点比率が 0.25 を下回り、対戦相手のエージェントのものよりも低かった回数である。

表 8 より、行動種類の変更のみを適用した場合は適用前後で平均獲得罰点比率が減少していることが分かる。一方で、SVR の適用のみを行った場合は適用前後で平均獲得罰点比率に差はあまり見られない。よって、行動種類を変更したことで、より応用的な行動選択規則を獲得することができるようになり、平均獲得罰点比率を減少させることができたと考えられる。しかし、それだけではモンテカルロエージェントに勝つことはできず、更に SVR を適用したことでモンテカルロエージェントに匹敵する学習結果が得られ、10 試行中 4 試行で、0.25 を下回った。これは、SVR を適用したことで学習中に選択される行動種類が変化し、より適した行動選択規則を獲得することができたためであると考えられる。よって、提案手法を適用することは、学習エージェントの平均獲得罰点比率を減少させる上で有効であると言える。

6.4 順位による比較

本節では、学習エージェントの平均獲得罰点比率ではなく、4 プレイヤ間の順位による評価を行う。実験方法としては、学習ゲーム数 N を 100,000 とし、ルールベースエージェント 3 体と対戦し学習させる。学習後、ルールベースエージェント 3 体及びモンテカルロエージェント 3 体と 1,000 ゲーム対戦させ平均順位を計測する。ここでは、学習エージェントを含めた 4 プレイヤの内、いずれかのプレイヤの合計獲得罰点が 100 点に達するまでを 1 ゲームとする。これを 10 試行行い、学習エージェントの平均順位を算出する。既存手法 [11] 及び提案手法に対する順位による評価を表 9 に示す。

表 9 より、ルールベースエージェント、モンテカルロエージェントの両エージェントに対して、提案手法を適用した学習エージェントの平均順位が既存手法 [11] のものより高いことが分かる。提案手法において、ルールベースエージェントに対しては、1 位率が 4 割近い値となっている。また、モンテカルロエージェントに対しては、平均順位の期待値が 2.5 に近い値となっており、モンテカルロエージェントとほぼ同等の性能にまで向上していることが分かる。よって、順位による評価からも提案手法の有効性を示していると言える。

表 7 学習ゲーム数が 100,000 の場合における学習後の学習エージェントの生成カテゴリ数

トリック	親	1 番目の子	2 番目の子	3 番目の子
1	4814	4183	4172	4037
2	2522	2463	2554	2232
3	1855	1605	1605	1290
4	1812	927	848	810
5	1274	766	786	813
6	868	803	831	845
7	701	829	851	859
8	787	807	780	768
9	1164	720	681	592
10	1633	593	563	525
11	2221	657	720	814
12	2935	1284	1407	1558
13	4234	2763	2892	3057

表 8 学習ゲーム数が 100,000 の場合における学習後の学習エージェントの平均獲得罰点比率

	ルールベース エージェントと 対戦した場合	モンテカルロ エージェントと 対戦した場合
先行研究	0.2395(10)	0.2714(0)
行動種類の変更のみ	0.2194(10)	0.2552(1)
SVR の適用のみ	0.2395(10)	0.2681(0)
両方を適用	0.2185(10)	0.2520(4)

表 9 学習ゲーム数が 100,000 の場合における学習後の学習エージェントの順位

Ranking	ルールベース エージェントと 対戦した場合		モンテカルロ エージェントと 対戦した場合	
	提案手法	既存手法	提案手法	既存手法
1	0.3821	0.2884	0.2327	0.1808
2	0.2805	0.2572	0.2570	0.2270
3	0.2095	0.2448	0.2551	0.2554
4	0.1279	0.2096	0.2552	0.3368
期待値	2.0832	2.3756	2.5328	2.7482

7. むすび

本研究では、ファジィ ART を拡張した機械学習手法である FALCON を用いたハーツ用の自律エージェントの性能を向上させるために、行動種類の改良手法と、SVR を適用した行動選択手法を提案した。また、提案手法の有用性を確認するために対戦実験を行った。

提案手法を適用することで、既存研究よりも学習エージェントの平均獲得罰点比率が減少することを確認した。特に、既存研究における学習エージェントはモンテカルロエージェントに勝つことができなかったが、本研究の提案手法を適用することにより、学習エージェントがモンテカルロエージェントを打ち負かすことがあることも確認することができた。また、ルールベースエージェントに対する平均獲得罰点比率を更に減少させることもできた。よって、提案手法は学習エージェントの性能

を向上させる上で有用だといえる。

今後の課題としては、SVR 学習モデルの分類性能を向上させることが挙げられる。訓練用データセットに用いる特徴や、LIBSVM のパラメータを調整することで、予測性能を向上させることができると考えられる。また本研究では、最も大きな罰点札である ♠Q に関する予測のみを行ったが、罰点札である ♠の各カードなど他の要素に関する予測も行うことで、より戦略的な行動選択規則を獲得することができると期待できる。さらに、対象問題の特徴を詳細に調査し、適切な回帰分析手法を検討することも今後の課題としてあげられる。

謝辞

本研究は広島市立大学特定研究（一般研究費）による支援を受けた。

参考文献

- [1] 星野准一: “特集「ゲーム AI」の企画にあたって,” 人工知能学会誌, Vol.23, No.1, p.43, 2008.
- [2] 松原仁: “Master の登場,” 情報処理, Vol.58, No.3, pp. 172-173, 2017.
- [3] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis: “Mastering the game of Go without human knowledge,” *Nature*, Vol.550, pp.354-359, 2017.
- [4] H. Ueda, N. Hanada, H. Kimoto, T. Naraki, K. Takahashi, and T. Miyahara: “Fuzzy Q-Learning with the modified fuzzy ART neural network,” *Intelligent Agent Technology, IEEE/WIC/ACM Int. Conf.*, pp. 308-315, 2005.
- [5] A. H. Tan: “FALCON: A Fusion Architecture for Learning, COgnition, and Navigation,” *Int. Joint Conf. on Neural Networks*, Vol.4, pp. 3297-3302, 2004.
- [6] H. Morishita, H. Ueda, and K. Takahashi: “Multi-agent reinforcement learning based on multi-channel ART networks,” *Int. Conf. Agents and Artificial Intelligence*, Vol.2, pp. 461-464, 2013.
- [7] D. Xiao and A. H. Tan: “Scaling up Multi-Agent Reinforcement Learning in Complex Domains,” *Int. Conf. on Web Intelligence and Intelligent Agent Technology*, Vol.2, pp. 326-329, 2008.
- [8] D. Wang, B. Subagdja, A. H. Tan, and G. W. Ng: “Creating human-like autonomous players in real-time first person shooter computer games,” *Twenty-First Innovative Applications of Artificial Intelligence Conf.*, pp. 173-178, 2009.
- [9] N. R. Sturtevant: “An Analysis of UCT in Multi-player Games,” *The 6th Int. Conf. on Computers and Games*, pp. 37-49, 2008.
- [10] 齋藤雄太, 鶴岡慶雅: “線形関数近似によるトリックテイキングゲームの Q 学習,” ゲームプログラミングワークショップ 2016 論文集, pp. 196-200, 2016.
- [11] K. Nimoto, K. Takahashi, and M. Inaba: “Improvement of Agent Learning for a Card Game based on Multi-channel ART Networks,” *J. OF COMPUTERS*, Vol.11, No.4, pp. 341-352, 2016.
- [12] K. Nimoto, K. Takahashi, and M. Inaba: “Construction of a Player Agent for a Card Game Using an Ensemble Method,” *Procedia Computer Science*, Vol.96, pp. 772-781, 2016.
- [13] C. C. Chang and C. J. Lin: “LIBSVM – A Library for Support Vector Machines,” <https://www.csie.ntu.edu.tw/~cjlin/libsvm/> [accessed Jul. 17, 2018]
- [14] L. Wolves: “Lone Wolves-Web, game, and open source development,” <https://launchpad.net/ubuntu/+source/gnome-hearts> [accessed Jul. 17, 2018]
- [15] L. Kocsis and C. Szepesvari: “Bandit Based Montecarlo Planning,” *European Conference on Machine Learning*, 2017.
- [16] V. N. Vapnik and A. Ya. Lerner: “Pattern recognition using gener-

alized portrait method,” *Automation and Remote Control*, Vol.24, 1963.

- [17] 小野田崇: “知の科学 サポートベクターマシン,” オーム社, 2008.
- [18] M. Whitley: “Mark’s Hearts Tips,” <http://mark.random-article.com/hearts/> [accessed Jul. 17, 2018]
- [19] n-let (えぬれつと): “ハーツ攻略,” <http://www51.atpages.jp/nlet/hearts/index.html> [accessed Jan. 27, 2017]

(2017 年 11 月 28 日 受付)

(2018 年 5 月 1 日 採録)

[問い合わせ先]

〒 731-3194 広島県広島市安佐南区大塚東 3-4-1

広島市立大学大学院

伊藤 崇

E-mail: ito@cm.info.hiroshima-cu.ac.jp

著者紹介

かさはら かずま
笠原 和真 [非会員]

2018 年広島市立大学大学院情報科学研究科博士前期課程修了。学生時代、エージェントの学習及び強化学習手法の研究に従事。

にもと けんた
二本 健太 [非会員]

2017 年広島市立大学大学院情報科学研究科博士前期課程修了。同年株式会社日立インフォメーションエンジニアリングへ入社、現在に至る。学生時代、エージェントの学習及び強化学習手法の研究に従事。

いとう たかし
伊藤 崇 [非会員]

2015 年広島市立大学大学院情報科学研究科博士前期課程修了。同年広島市立大学大学院情報科学研究科博士後期課程進学、現在に至る。エージェントの学習及び遺伝的手法の研究に従事。電子情報通信学会会員。

たかはし けんいち
高橋 健一 [正会員]

1979 年名古屋工業大学大学院工学研究科修士課程了。同年名古屋工業大学工学部助手。同大講師、助教授を経て、1994 年広島市立大学情報科学部教授。現在同大学院情報科学部研究科所属。知識処理、エージェント、eラーニング等の研究に従事。工学博士。IEEE、電子情報通信学会、情報処理学会、日本知能情報ファジィ学会、各会員。

いしば みちまさ
稲葉 通将 [非会員]

2012 年 名古屋大学大学院情報科学研究科博士後期課程修了, 同年広島市立大学大学院情報科学研究科助教, 現在に至る. 対話システム, 対話処理に関する研究に従事. (博士 (情報科学)). 電子情報通信学会, 情報処理学会, 人工知能学会各会員.

Learning Experiments of a FALCON Using SVR for a Card Game

by

Kazuma KASAHARA, Kenta NIMOTO, Takashi ITO, Kenichi TAKAHASHI, and Michimasa INABA

Abstract:

This paper proposes a method to improve the learning performance of a learning agent with FALCON, to make a player agent for the card game “Hearts”, which is one of the multi-player imperfect-information games. FALCON is a machine learning method which is an extended fuzzy ART (Adaptive Resonance Theory). The previous work showed that FALCON is effective for Hearts. In this study, to improve the learning performance, the action set of the agent is changed based on strategies of Hearts, and a method that employs a prediction by the support vector regression is proposed.

Keywords: reinforcement learning, fuzzy ART, FALCON, multi-player imperfect-information game, support vector regression

Contact Address: **Takashi ITO**

*Graduate School of Information Sciences, Hiroshima City University
3-4-1 Ozukahigashi, Asaminami-ku, Hiroshima-shi, Hiroshima 731-3194, Japan.
E-mail: ito@cm.info.hiroshima-cu.ac.jp*