

RoboCup 小型リーグにおける行動系列に基づく相手戦略の学習

Learning an opponent strategy based on a sequence of actions in RoboCup SSL

安達勇介, 伊藤正英, 成瀬正

Yusuke Adachi, Masahide Ito and Tadashi Naruse

愛知県立大学 情報科学部

School of Information Science and Technology, Aichi Prefectural University

is121004@cis.aichi-pu.ac.jp, {masa-ito,naruse}@ist.aichi-pu.ac.jp

Abstract

RoboCup Soccer 小型リーグでは, 近年サッカー戦略が著しく進歩しており, 相手の行動を予測した上で対策行動を取ることが重要になってきている. このことは, 相手行動の学習という問題の解決を図る必要性を示唆している. 我々は, セットプレーを対象にして, ロボットの動き (配置) の類似性から相手戦略の分類を行う手法を検討してきた [2]. この手法には, セットプレーというトリガ点を与えること, および, (セットプレー後の) プレー時間が長い場合に処理時間が長くなる, という問題点がある. そこで本論文では, これらの制約を取り払うために, ロボットの一連の動作をキックやマークなどの基本行動の系列で表し, これを用いた相手戦略の分類を試みる. 分類結果と正解データを比較した結果, 我がチーム RoboDragons に対しては 4 試合中 3 試合において Rand Index が 0.878 を超え, 相手の 4 チームに対しては 0.802 以上の高い分類性能を得ることができた.

1 はじめに

近年, ロボットの動きを指令する戦略プログラム (以下, 戦略) は高度化しており, 相手の戦略を予測して対処することが重要課題となっている. 相手の戦略を学習しようという試みは, セットプレー戦略に関しては Erdogan ら [1] や 安井ら [2] の研究で行われている. しかしこれらはロボットの位置座標を元にした連続データのみを使用しており, 学習後の対策行動を直接求められないことや, ノイズへの弱さ, ロボットの軌道の照合の関係上 1 回のプレー (セットプレーからインプレーに移行し, 再びセットプレーが行われるまでの間) 全体へ範囲拡張することの困難さがあげられる.

本研究では, これらの課題を解決するため, 新たにロボットの行動を単位とし, 行動の系列によって戦略を分類することを試みる. 本手法は, 細かい座標データの動きにとらわれずにより大きな概念から分類を行い, また過去の行動連続データとの比較で, 次に来る行動そのものを予測することで対策行動を立てやすくする利点がある. 以下, 本論文では, ロボットの行動の抽出法について述べる. 次いで, 行動系列の非類似度を定義し, 非類似度計算アルゴリズムおよびクラスタリング手法を示す. 最後に, 実験結果を示し, 提案手法の有効性を示す.

2 関連研究

Erdgan ら [1] は, セットプレーの開始からボールが外に出るか相手に取られるまでの間の攻撃ロボットの軌道を取得し, これらを用いてセットプレー戦略の分類を行った. 一方, 安井ら [2] はセットプレー開始までのロボットの配置の連続データを使用してセットプレー戦略の分類を行っている.

本論文で提案する手法は, 新たにロボットの行動を使用してセットプレー及び任意の範囲でのプレーの分類を試みるものである.

ここで, ロボットの行動の検出に関しては浅野ら [3] や 安井ら [4] による先行研究が存在する. 浅野らはキックされたボールの直線運動を検出し, これの目的を判別すること, 及びマーク (パス妨害型マーク, シュート妨害型マーク, ボール保持者へのマーク) を検出することを行っている. 安井らは浅野らのボールの直線運動検出法の改善案を示した. 本論文ではこれらの手法を継承しつつ, 今回の目的に合わせるための修正, 機能追加を行い, 行動抽出を行う.

3 ロボットの行動の抽出方法

小型ロボットリーグでは, 相手の行動を分類する手法が浅野ら [3] によって, キック行動の検出が安井ら [4] によって

提唱されている。本研究ではこれらの手法を継承しつつ、現在の試合状況に則した手法へと改良を行った。

3.1 マーク行動の抽出

浅野らはマークの行動をパス妨害型マーク、シュート妨害型マーク、ボール保持へのマークの3種類に分類し、それぞれの検出方法を示した。ここでは浅野らの手法の紹介と、変更を加えた部分を示す。

3.1.1 浅野らによるマーク検出方法

以下に浅野ら [3] の手法を示す。

- $R, (x_i, y_i)$: 味方ロボット R と時刻 i での位置座標
- $R_j, (x_{ij}, y_{ij})$: 相手ロボットと時刻 i で位置座標
- $L : ax + by + c = 0$, パス妨害型マークの場合, ボールと味方ロボット R を結ぶ線. シュート妨害型マークの場合, ロボット R と相手ゴール中心を結ぶ線. ボール保持者へのマークの場合, R をボール保持者として, R とボールを結ぶ線.
- T : しきい値

として, 次のアルゴリズムを実行する.

Step1 $i = 1$ とする.

Step2 各 j に対して次の式を計算する. $E1$ は敵ロボットと味方ロボットとの距離である.

$$E1_{ij} = \sqrt{(x_i - x_{ij})^2 + (y_i - y_{ij})^2} \quad (1)$$

相手ロボット R_j から L に下ろした垂線が L に交わるかどうかで, R_j と L との距離を表す $E2$ の値を以下のように決定する.

$$E2_{ij} = \begin{cases} \frac{|ax_{ij} + by_{ij} + c|}{\sqrt{a^2 + b^2}} & (\text{垂線と } L \text{ が交わる}) \\ \frac{|ax_{ij} + by_{ij} + c|}{\sqrt{a^2 + b^2}} \cdot \gamma & (\text{垂線と } L \text{ が交わらない}) \end{cases} \quad (2)$$

2つの評価値を重み α, β を使用して足し合わせたものを E_{ij} とする.

$$E_{ij} = \alpha \cdot E1_{ij} + \beta \cdot E2_{ij} \quad (3)$$

この E_{ij} の n フレーム平均を取る.

$$\bar{E}_j = \frac{1}{n} \sum_{k=i}^{i+n-1} E_{kj} \quad (4)$$

$$CoveringByRobot_j = \begin{cases} 0 & (\bar{E}_j > T) \\ 1 & (\bar{E}_j \leq T) \end{cases} \quad (5)$$

もし $CoveringByRobot_j = 1$ であった場合, i から $i+n-1$ の間 R は R_j にマークをされたとする. $i = i+n$ として, 試合ログデータの終わりまで Step2 を繰り返す.

3.1.2 パス妨害型マーク

浅野らの手法ではパス相手の存在の有無が考慮されておらず, 単にボールからロボットへのコース上を妨害しているかどうかの評価式であった. ボールを追っているだけの状態のようなケースを除外し, より忠実にパス妨害を判定できるようにしたものが次のアルゴリズムである.

$T_{Nearest}$ はパス相手であり, そのフレームにおいて, マーク対象となっているロボット以外のうち, 直線 L との距離が D 未満¹ である中で, 最小距離の味方ロボットとする.

- $\vec{T}_i = (x_{ti}, y_{ti}) (i = \{0, 1, \dots, 5\})$: フレーム f における味方ロボット T_i の位置ベクトル
- $\vec{O}_j = (x_{oj}, y_{oj}) (j = \{0, 1, \dots, 5\})$: フレーム f における敵ロボット O_j の位置ベクトル
- \vec{B} : ボールの位置ベクトル
- $\vec{V}_1 = \vec{B} - \vec{T}_i$: T_i からボールへのベクトル
- $\vec{V}_2 = \vec{O}_j - \vec{T}_i$: T_i から O_j へのベクトル
- $\vec{T}_{Nearest} = (x_n, y_n)$: フレーム f で, \vec{V}_1 の延長線上に最も近い T_i 以外の味方の位置ベクトル
- $\vec{V}_3 = \vec{T}_{Nearest} - \vec{T}_i$: T_i から $T_{Nearest}$ へのベクトル
- $\vec{V}_4 = \vec{T}_i - \vec{T}_{Nearest}$: $T_{Nearest}$ から T_i へのベクトル
- $\vec{V}_5 = \vec{O}_j - \vec{T}_{Nearest}$: $T_{Nearest}$ から O_j へのベクトル

Step0 フレーム番号 $f, j = 0, i = 0$ とする.

Step1 T_i と O_j に対して以下の式を計算する

$$Dot_1 = \frac{\vec{V}_1}{|\vec{V}_1|} \cdot \vec{V}_2 \quad (6)$$

$$Dot_2 = \frac{\vec{V}_4}{|\vec{V}_4|} \cdot \vec{V}_5 \quad (7)$$

$$Val_{fj} = \begin{cases} \left| \frac{\vec{V}_3}{|\vec{V}_3|} \times \vec{V}_2 \right| & (Dot_1 \geq 0 \wedge Dot_2 \geq 0) \\ \gamma_p + \left(\left| \frac{\vec{V}_3}{|\vec{V}_3|} \times \vec{V}_2 \right| \right) & (Dot_1 < 0 \vee Dot_2 < 0) \end{cases} \quad (8)$$

Val_{fj} は味方ロボット T_i と $T_{Nearest}$ を結ぶ線分に O_j の中心から下ろした垂線の長さである. 味方ロボット2台よりも外側に存在する場合, それはパス経路の妨害をしていないことになるので, 重みを加算して除外する.

$$MarkPass_j = \begin{cases} 0 & \left(\frac{1}{n} \sum_{k=f}^{f+n-1} Val_{kj} > T_p \right) \\ 1 & \left(\frac{1}{n} \sum_{k=f}^{f+n-1} Val_{kj} \leq T_p \right) \end{cases} \quad (9)$$

¹今回は 400(mm) を使用した.

評価平均値が閾値 T_p よりも小さければ、 T_i は f から n フレームの間は敵ロボット O_j によってマーク行動が行われたとし、行動を MarkPass とする²。

Step2 i をインクリメントする。もし i が 5 ならば、 j をインクリメントし、 $i = 0$ として Step1 に戻る。 j が 5 ならば、このフレームでのマーク計算は終了である。

3.1.3 シュート妨害型マーク

シュート妨害型マークは、浅野らの手法ではボール保持者との距離が評価式に含まれていたが、シュート妨害は離れた位置で行われることが多いためこれを削除した。具体的には式 3 の $\alpha = 0, \beta = 1$ とし、また修正したパス妨害型マークと同様に垂線が L に交わらない場合は完全に除外するように設計した。シュート妨害型マークの状態のとき、行動を MarkShoot とする³。

3.1.4 ボール保持者へのマーク

ボール保持者へのマーク行動検出は、ボールの後ろやロボットの後ろから接近している場合もボール保持者へのマークと考えられるため、式 2 を $\gamma = 1.0$ とした。つまり、位置関係による除外を行わない。ボール保持者へのマークの状態であるとき、行動を MarkBall とする⁴。

3.2 パス待ち行動の抽出

今回新たにパス待ちの検出を設計した。基本的に n 回の平均を取るアイデアは浅野によるものを採用し、パス待ちという行動の評価値を新たに設計したことになる。

Step0 最もボールに近い敵ロボットを探索し、 O_{ball} とする。

Step1 フレーム k で、 O_{ball} 以外の敵ロボット O_j について、もし O_{ball} であるか、相手の陣地奥に存在する場合、空き角度 θ_{kj} に負の定数を入れ、パス待ちの判定から除外する。

そうでない場合、 O_j から見た自ゴールに対しての空き角度 θ_{kj} - 障害物がおらず、シュートが狙える空き角度 - を計算する。

Step2 以下のように n フレーム間での平均値を取り⁵、それがしきい値 T_w よりも大きい場合、そのロボットはパス待ちをしているとし、行動を WaitPass とする⁶。

$$WaitPass_j = \begin{cases} 1 & \left(\frac{1}{n} \sum_{k=f}^{f+n-1} \theta_{kj} \geq T_w \right) \\ 0 & \left(\frac{1}{n} \sum_{k=f}^{f+n-1} \theta_{kj} < T_w \right) \end{cases} \quad (10)$$

²実験では $T_p = 400(\text{mm})$ を使用した。

³実験ではしきい値 $T = 400(\text{mm})$ を使用した。

⁴重み $\alpha = \beta = 1/2$ 、しきい値 $T = 400(\text{mm})$ を使用した。

⁵平滑化フレーム数はマークとパス待ち検出で $n = 3$ を使用した。

⁶今回の実験では $T_w = 0.139626(\text{rad})(8 \text{ 度})$ を使用した。

3.3 キック行動抽出とキックの種類判別

ロボカップサッカーにおいて、マーク行動とともに重要なものがキック行動である。キックそのものの検出については安井ら [4] が行っているため、本研究ではこれによって検出されたキックの種類を判別する仕組みを新たに設計した。浅野ら [3] が行ったキック結果判別と目的とするところは類似しているが、ここでは目的にのみ注目し、シュート、パス、クリアのみをキックの種類とする。また、キックの延長線とオブジェクトとの接触だけではなく、近年の試合に多く見られるゴール近辺でのシュート、パスの分類ができるよう設計を行った。アルゴリズムを次に示す。

Step0 キック行動の種類 Kick を、

Kick = {KickBall, KickShoot, KickPass, KickClear} とする。KickBall はシュート、パス、クリアのどれにも該当しないキックであり、KickShoot はシュート、KickPass はパス、KickClear はクリアを目的としたキックである。

Step1 キック開始地点とボールの直線運動終了地点を結ぶベクトル \vec{P}_b を求める。

Step2 \vec{P}_b の終点の座標がフィールドのタッチライン (サイドライン) よりも外であった場合、クリア (KickClear) とする。

Step3 \vec{P}_b の終点の座標が自チームのゴールラインよりも外であった場合、シュート (KickShoot) とする。

Step4 キック開始点から他の相手ロボットへのベクトル \vec{P}_{oi} を求め、

$$\left| \vec{P}_{oi} \times \vec{P}_b \right| < D_1 \quad (11)$$

を満たしている場合、パス (KickPass) とする⁷。

Step5 キック開始点から、自チームのゴール右端と左端にマージンを加えた地点へのベクトル $\vec{P}_{gr}, \vec{P}_{gl}$ を用意する。キックの始点から終点へのベクトルの角度が、 $\vec{P}_{gr}, \vec{P}_{gl}$ の角度の間に収まる場合、シュートの方向に蹴られたことがわかる。このとき、 \vec{P}_b の終点の座標と自分のゴール中心との距離 d がしきい値 D_G よりも小さい場合、ゴールを狙った途中で何かにあたったと考えられるので、これはシュート (KickShoot) とする⁸。

Step6 どの目的にも該当しない場合、KickBall とする。

4 行動抽出の全体の流れ

これまでの行動抽出アルゴリズムを次のように動作させる。

⁷本研究では $D_1 = 800(\text{mm})$ を使用した。

⁸ $D_G = 2000(\text{mm})$ を実験では使用している。

Step1 現在のロボットやボールの位置が保存されたフレームデータを取得する。

Step2 キック検出，パス待ち検出，マーク検出を行う。

Step3 各ロボットについて，検出された行動をキューに保存していく．検出された行動が以前に現れた行動の続きである場合，その行動を保存した場所のデータを更新する．Step1 に戻る．

行動は，現れた順に保存されていく．優先度は高いものからキック，パス待ち，マークの順としており，上位の行動が継続している間は下位の行動は検出されず，下位の行動継続中に上位の行動が検出された場合はその下位の行動は終了し，上位の行動に切り替わる．

4.1 得られた行動系列に対する前処理

行動系列が得られた段階では，ノイズと考えられる行動データも含まれている．これらを次のように取り除く．

- 数フレーム⁹で切り替わった行動はノイズと判断するべきで，このような行動に対しては除去を行う．ただし，キックの場合はすぐにフィールド外に出たような場合も重要な特徴となり得るため，前後にキックの行動がない場合はこの処理は行わない．
- 同じ行動が途切れ，再開された場合，これは同一の行動として扱うべきであるので，統合する処理を行う．ただし，キックの場合，連続してキックするということは重要な意味を持つので，一定フレームの差¹⁰がなく同一と考えられる場合のみ統合を行う．
- 特に短い範囲で行動抽出をした場合，行動が1つも抽出できない場合があるが，その場合に限り行動 No-Data を追加する．

5 行動の非類似度

前章までの行動抽出法により，シュートマーク，パスマーク，ボール保持者へのマーク，パス待ち，キック，パス，シュート，クリアの8種類 (NoData を含めると9種類) の行動が抽出できることを示した．この行動の種類を $action$ とし，特にキック4種類を Kick グループと定義する．試合中にはこれらの開始点，終了点，継続フレーム数を加えて，次の形で行動データが保存されていく．このとき， P 回目のプレーにおいて，6 台のロボットのうち n 番目のロボットに対して t 回行動抽出した行動系列 $A_P[n]$ は，行動開始点 \vec{p}_{sn1} ，行動終了点 \vec{p}_{en1} と継続フレーム数 $frame$ を合わせて，

次のような形となる．

$$A_P[n] = \left[\begin{array}{c} action_{n1} \\ \vec{p}_{sn1} \\ \vec{p}_{en1} \\ frame_{n1} \end{array} \right], \dots, \left[\begin{array}{c} action_{nt} \\ \vec{p}_{snt} \\ \vec{p}_{ent} \\ frame_{nt} \end{array} \right] \quad (12)$$

この系列のうち， k 番目の1要素を取り出したものを $A_P[n][k]$ と定義する．

$$A_P[n][k] = \left[\begin{array}{c} action_{nk} \\ \vec{p}_{snk} \\ \vec{p}_{enk} \\ frame_{nk} \end{array} \right] \quad (13)$$

2つの行動 $A_{P_1}[n_1][t_1]$, $A_{P_2}[n_2][t_2]$ の非類似度 d_0 を，重み α, β, γ を用いて，次のように定義する．それぞれの項は行動の種類的一致，位置的一致，系列の長さの一致を表している．行動の種類が一致している場合，行動継続期間の全体に占める割合で一致具合を表現する．Kick は重要な行動であるため，Kick と他の行動のマッチングに対しては距離が離れるように設計した．

$$d_0(A_{P_1}[n_1][t_1], A_{P_2}[n_2][t_2]) = \begin{cases} \alpha \cdot \text{frame_diff} + \beta \cdot \text{p_distance} + \gamma \cdot \text{diff_size_cost} & (action_{n_1 t_1} = action_{n_2 t_2}) \\ \alpha \cdot 2.0 + \beta \cdot \text{p_distance} + \gamma \cdot \text{diff_size_cost} & (action_{n_1 t_1} \in \text{Kick} \wedge action_{n_2 t_2} \notin \text{Kick}) \\ \alpha \cdot 1.0 + \beta \cdot \text{p_distance} + \gamma \cdot \text{diff_size_cost} & (action_{n_1 t_1} \notin \text{Kick} \wedge action_{n_2 t_2} \in \text{Kick}) \\ \alpha \cdot 1.0 + \beta \cdot \text{p_distance} + \gamma \cdot \text{diff_size_cost} & (\text{otherwise}) \end{cases} \quad (14)$$

frame_diff は，注目する行動が行われたプレーのフレーム数 $frame_play$ に対して，注目する行動がどれだけの割合を占めるかの非類似性を表す評価値である．0 から1までの値を取る．

$$\text{frame_diff} = \left| \frac{frame_{n_1 t_1}}{frame_play_{n_1 t_1}} - \frac{frame_{n_2 t_2}}{frame_play_{n_2 t_2}} \right| \quad (15)$$

p_distance は，開始位置と終了位置の距離の0から2の評価値を表し，次の式で定義する． $FieldLength$ はフィールド長手方向の長さである． x 軸 (自ゴール中心から相手ゴール中心へ正の軸) に対して座標の対称性を考慮するため， y 座標 (y 軸は x 軸の90度左回転) を反転させたものと，そうでないものとで値が小さい方を採用する．

$$\begin{aligned} \text{p_distance} &= \min \left\{ \frac{|(\vec{p}_{sn_1 t_1} - \vec{p}_{sn_2 t_2})|}{FieldLength}, 1.0 \right\} \\ &+ \min \left\{ \frac{|(\vec{p}_{en_1 t_1} - \vec{p}_{en_2 t_2})|}{FieldLength}, 1.0 \right\} \end{aligned} \quad (16)$$

⁹今回は6フレーム (0.1s) とした．

¹⁰今回は60フレーム (1.0s) と設定した．

diff_size_cost は、対象とする行動が含まれる行動系列の大きさの差を追加コストとして与える評価値である。最小は 0 であり、1:4 の差がある場合に 1 を取り、1:7 以上の差がある場合に 2 を取る。長い方の系列の大きさを long_size、短い方を small_size とすると次のようになる。

$$\text{diff_size_cost} = \min \left\{ \frac{1}{3} \left(\frac{\text{long_size}}{\text{small_size}} - 1.0 \right), 2.0 \right\} \quad (17)$$

ここで、あるロボット n_1 とあるロボット n_2 の行動系列全体 $A_{P_1}[n_1], A_{P_2}[n_2]$ 同士の非類似度 $d_1(A_{P_1}[n_1], A_{P_2}[n_2])$ を定義する。行動系列の長さは同じとは限らないため、短い方を基準とし、長い方の中にどれだけ中身が一致するかを調べ非類似度とする。系列と系列の非類似度 d_1 は次のアルゴリズムにより決定する。

Step1 系列の長さを比較し、短い方を small、長い方を long とする。それぞれの長さは small_size、long_size と表現する。long の中に含まれる行動のうち Kick の数をカウントし、これを *kick_num* とする。small と long の探索用カウンタを i, j とし、初期値は 1、long の探索開始点を $start_j = 1$ 、long の探索終了点 $limit_j = 1$ とする。 $d_1 = 0$ として Step2 を開始する。

Step2 まず、small で選択中の i 番目の行動に対して、long 中の探索範囲を決定する。これは長さの比を元に次のように決定する。なお、小数点以下は切り捨てを行う。

$$\begin{aligned} ls &= \text{long_size}/\text{small_size} \\ limit_j1 &= i + ls \\ limit_j2 &= \min(start_j + ls, \text{long_size}) \\ limit_j &= \max(limit_j1, limit_j2) \end{aligned} \quad (18)$$

small の i 番目の行動と、long の $start_j$ から $limit_j$ までの行動とで、行動の種類 *action* が一致するまで探索を行う。

Step3 一致する行動が存在した場合、

$$d_1 = d_1 + d_0(A_{P_1}[n_1][i], A_{P_2}[n_2][j])$$

を計算し、 $start_j = j + 1$ とする。一致する行動が存在しない場合、

$$d_1 = d_1 + d_0(A_{P_1}[n_1][i], A_{P_2}[n_2][i])$$

の計算を行う。 $i < \text{small_size}$ の場合は $i = i + 1$ として Step2 へ、そうでない場合 Step4 に進む。

Step4 最後に、Kick の数 *kick_num* のうち使用されずに残った数 *kick_unuse* を追加コストとして加算する。

$$d_1 = d_1 + \text{kick_unuse}$$

最後に、あるプレーで得られたロボット 6 台分の行動系列と、別のプレーで得られたロボット 6 台分の行動系列の対応関係を加味してプレー同士の非類似度 d_2 とする。 P_σ は置換行列である。

$$d_2(A_{P_1}, A_{P_2}) = \min_{\sigma \in S_6} \{ \text{Tr}(FP_\sigma) \} \quad (19)$$

$$F = [f_{ij}] \quad (20)$$

$$f_{ij} = \{d_1(A_{P_1}[i], A_{P_2}[j])\} \quad (21)$$

ここで定義した類似度を用いて群平均法 [5] でクラスタリングを行う。

6 クラスター数の決定

本研究では、安井ら [2] による研究で用いられた手法によりクラスター数を決定する。この手法は、あらかじめクラスター数の範囲を指定する必要がある Davies Bouldin index[6] などとは異なり、範囲指定する必要がない。相手戦略といったクラスター数が不明なものを分類する場合に有効な手法である。安井らが提案した手法は次の通りである。

X_k を k 回目のプレー、 A_{P_k} をそのときの行動系列として、

$$W(K) = \sum_{i=1}^K \sum_{X_k \in C_i} \sum_{X_l \in C_i} d_2(A_{P_k}, A_{P_l}) \quad (22)$$

を定義する。この式は、 K 個のクラスターに分類されたときの、それぞれのクラスター内の要素（プレー）同士の距離の総和を表している。つまり、クラスター内の他の要素と比べて非類似度の高い（距離の遠い）要素がクラスター内に存在する場合、この評価値は要素間の距離の和を取るため、値が大きくなる仕組みとなっている。この $W(K)$ を用いて、

$$W'(K) = W(K)/W(1) \quad (23)$$

としたとき、

$$\arg \max_{1 \leq K \leq N} (W'(K) \leq h) \quad (24)$$

となるようなクラスター数 K を採用する。

7 試合ログデータを用いた実験結果と考察

RoboCup2015 で我々のチーム RoboDragons の行った公式戦 4 試合について、RoboDragons の行動がどのように分類されるか実験を行った。この実験では非類似度の重みを $\alpha = \beta = \gamma = 1/3$ とした。クラスター数決定のしきい値 h については、0.06 を使用する。この値は先行研究 [2] で最も良い結果が出たときの値である。

今回は人の目で行動の分類の行いやすいセットプレーを対象にする。ここでのセットプレーは、ボールの再配置から、ボールが相手に取られるかフィールド外に出るまでの間とする。それぞれの試合についてクラスタリングを

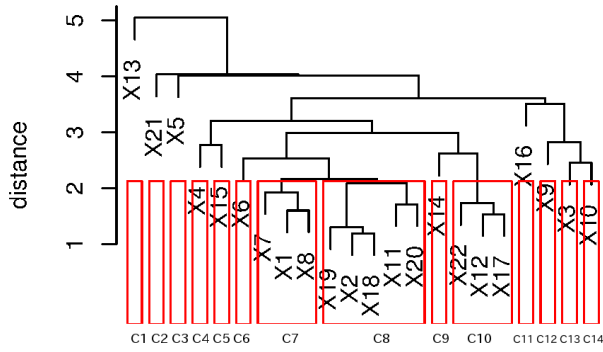


Figure 1: Dendrogram for match No.1
(group average clustering; the same clustering is used in the following figures)

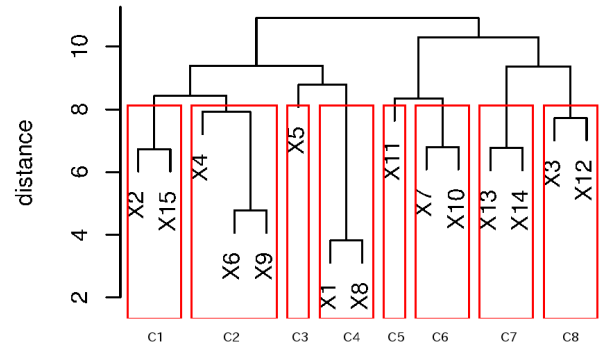


Figure 3: Dendrogram for match No.3

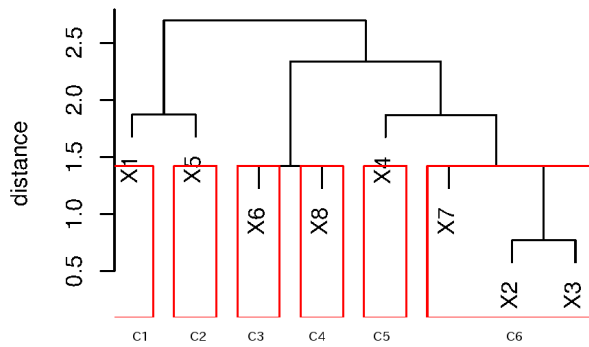


Figure 2: Dendrogram for match No.2

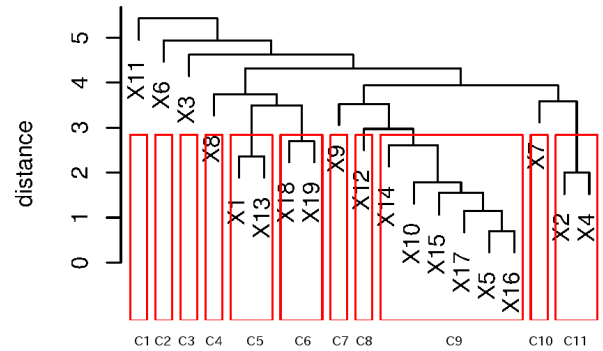


Figure 4: Dendrogram for match No.4

行った結果が図 1 から図 4 である。これら分類結果と、筆者が行動に着目して行った分類結果とを Rand Index[7] で評価を行った。その結果が表 1 である。第二試合の結果を除くと正解データとした筆者の分類結果と高い水準で類似していることがわかる。第二試合に関して調査したところ、相手が不調により動かなかった場合があり、マーク行動の判定が消滅したことによる分類悪化が起きていることがわかった。0.878 や 0.879 という水準で止まった試合については、正解データとして用意したクラスターが提案手法では 2 つに分離される現象が起きており、これが大きな原因と考えられる。実用上では同じ行動が 2 つのクラスターに分かれていても問題は少ないため、十分役立つのではないかと考えている。

Table 1: Rand Index (RoboDragons)
(computer clustering vs. human clustering)

	No.1	No.2	No.3	No.4
Rand Index	0.879	0.750	0.990	0.878

8 他のチームでの結果

RoboDragons の場合を前章で見たが、公式戦 4 試合の相手チームに対して適用した場合どうなるのかを本章で示す。結果が表 2 であり、各々のクラスタリング結果が図 5 から図 8 である。4 試合とも Rand Index は 0.80 以上の値となった。今回の結果も自チームのときと同様に、正解として用意したクラスターが 2 つ以上に分裂する現象が起きていることがわかった。この問題を解決するには、類似したプレーを分離させずにまとめる手法を考案することが挙げられ、現状の郡平均法を使用したクラスタリング部を再考する必要があると思われる。例えば、最短距離法で一定距離までを結合する、などが考えられるが、今後検討を進める必要がある。

9 実行時間

最も多い 35 回のセットプレーが行われた 4 戦目の対戦相手の場合について、クラスタリングの部分における実行時

Table 2: Rand Index (opponents)

	No.1	No.2	No.3	No.4
Rand Index	0.853	0.861	0.850	0.802

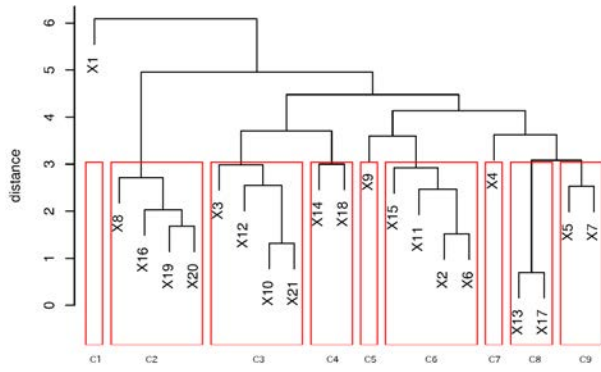


Figure 5: Dendrogram for match No.1(opponent)

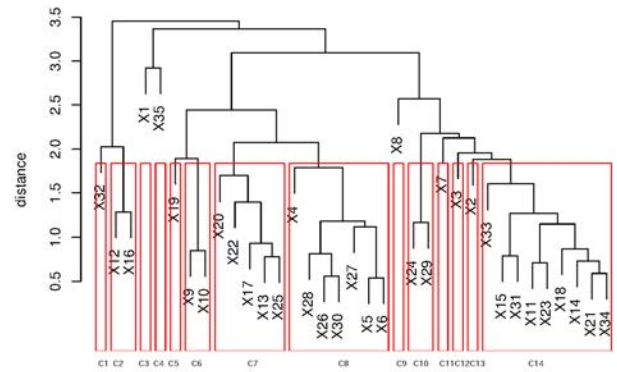


Figure 8: Dendrogram for match No.4(opponent)

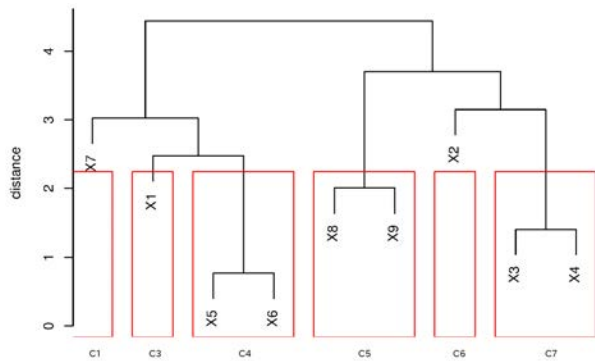


Figure 6: Dendrogram for match No.2(opponent)

間の計測を行った。この部分では行動系列の前処理を行い、距離行列の作成と、郡平均法で全て単独クラスターの状態から1つのクラスターになるまで結合を繰り返す処理を行う。(1回目では単独クラスター1つしかないのでそのまま終わり,35回目では35回目と,今までの34回分のプレー全てと距離の計算を行い,その後クラスタリングを行う。)この結果が表3であり,この結果から十分リアルタイム性を持って計算が行えることがわかる。

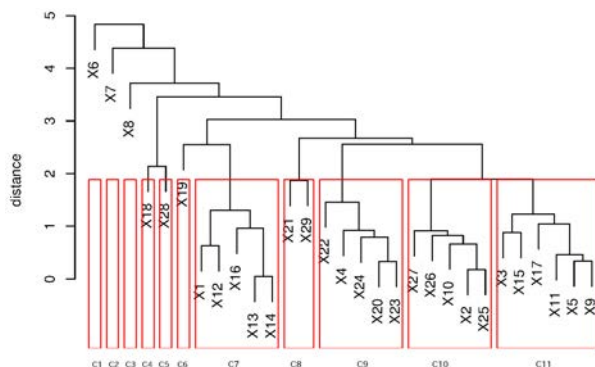


Figure 7: Dendrogram for match No.3(opponent)

Table 3: clustering time(AMD A10 7800)

	average time(ms)	max time(ms)
No.4(35 setplays)	0.67	1.82

10 おわりに

本研究では新たに行動を単位として相手の戦略を学習する仕組みを設計した。正解データとの間で Rand Index を計算すると RoboDragons に対して1試合の例外を除いて0.878以上,相手チームに対しては0.802以上の評価値が得られ,十分正解に近い結果が得られていると言える。実行時間に関しても実時間で計算できることを示した。今後は高精度化及び,より広いプレー範囲での検証や,過去の行動系列データとの照合による対策行動の生成,自チームの戦略へのリアルタイム反映といった課題に取り組んでいきたい。

参考文献

- [1] Can Erdogan, et al. "Action Selection via Learning Behavior Patterns in Multi-Robot Domains", IJCAI,2011
- [2] 安井興太郎, 他, RoboCup 小型リーグにおけるリアルタイム学習のための相手行動の分類, 電子情報通信学会 論文誌 VOL. J97-D NO.8, 2014
- [3] Koshi Asano, et al. "Detection of basic behaviors in logged data in RoboCup Small Size League", RoboCup Symposium , 2008
- [4] 安井興太郎, 他, RoboCup 小型ロボットリーグにおけるキック検出手法の検討, 人工知能学会 AI チャレンジ研究会 SIG-Challenge,2012
- [5] B.S.Everitt et al. "Cluster Analysis 5th Edition" , Wiley, 2011
- [6] David L. Davies and Donald W. Bouldin, "A Cluster Separation Measure" , IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI- 1(2), pp.224-227, 1979
- [7] W. M. Rand, "Objective criteria for the evaluation of clustering methods", Journal of the American Statistical Association (American Statistical Association) 66 (336): 846-850,1971