

# ランキング学習を用いたサッカーエージェントの行動評価関数の獲得

秋山 英久<sup>a)</sup> 辻 将司 荒牧 重登<sup>b)</sup>

概要：本研究では，サッカーシミュレーションにおける選手エージェントの行動評価関数を，ランキング学習を用いて獲得する手法を提案する．実装では，選手エージェントのボールキック行動時の意思決定に利用されている，行動連鎖探索フレームワークを利用する．この行動連鎖探索フレームワーク中の評価関数をランキング SVM によって獲得させる．

## 1. はじめに

近年の計算機能力の向上に伴い，サッカーのような連続状態行動空間かつリアルタイムの意思決定が求められるタスクにおいても，オンラインでの探索手法が適用されるようになってきた．しかしながら，探索において生成された行動を評価する適切な行動評価関数は明らかではなく，多くの場合，人手によるプログラミングで行動評価関数が設計されている．チームの戦略や戦術を反映した行動評価関数を設計することは容易ではなく，行動評価関数に含まれるパラメータ調整にも多大な労力が要求されている．本研究の目標は，サッカーシミュレーションにおける選手エージェント（以下，サッカーエージェントと呼ぶ）の行動評価関数をより直感的，柔軟に設計可能とすることである．本稿では，ランキング学習を用いてサッカーエージェントの行動評価関数を獲得する手法を提案する．

## 2. 関連研究

マルチエージェントシステム研究のテストベッドとして，RoboCup サッカー 2D シミュレータ（以下 RCSS と呼ぶ）が知られている [7]．RCSS 環境において機械学習を適用する取り組みとしては，サッカーエージェントの協調的な意思決定あるいは最適な個体制御を，強化学習によって獲得させるアプローチが多い．Stone らは強化学習のテストベッドとして Keepaway というサッカーのサブタスクを提案している [10]．また，Gabel らは敵エージェントの攻撃の行動に対する 1 対 1 の守備タスクを設定し，実用的

な性能を持った意思決定能力を強化学習によって獲得することに成功している [1]．これらの研究では，比較的小規模なタスクに限定することで強化学習の適用が成功している．しかしながら，11 対 11 の試合において協調行動を獲得するには至っていない．

従来，サッカーエージェントの協調的な振る舞いは事前に人手で設計し，あらかじめプログラムへ組み込んでおくアプローチが一般的であった [5], [8], [9]．これに対して，計算機の性能が向上したことで，サッカーエージェントが取りうる行動列の候補をその場で列挙し，それら进行评估して選択するアプローチが可能となりつつある．秋山らはサッカーエージェントのボールキック行動に対して木探索を適用し，複数のサッカーエージェントによる行動列をオンラインでプランニングするフレームワークを提案している [4]．このフレームワークでは，サッカーエージェントが生成・列挙した行動に評価値を付与し，探索終了後，サッカーエージェントはその値がもっとも高い行動を実行する．しかしながら，探索中に生成された協調行動を評価するための行動評価関数は人手で設計されている．そのため，チームの戦略や戦術を反映するための特徴の選択，そして，それらに関するパラメータ調整にも多大な労力が要求されている．

## 3. サッカーエージェントへのランキング学習の適用

本節では，まず [4] の行動連鎖探索フレームワークについて説明する．次に，このフレームワーク上でのランキング学習の適用方法について説明する．

<sup>1</sup> 福岡大学  
8-19-1 Nanakuma, Jonan-ku, Fukuoka 814-0180, Japan

<sup>a)</sup> akym@fukuoka-u.ac.jp

<sup>b)</sup> aramaki@fukuoka-u.ac.jp

### 3.1 行動連鎖探索フレームワーク

行動連鎖探索フレームワークは、自分と他者を含めた複数のエージェントによって実行される行動（サッカーの場合はパス、ドリブル、シュートなど）を生成し、探索木にノードとして格納していくことで有効な行動連鎖の探索を実行する。サッカーにおける行動連鎖のイメージを図1に示す。この図では、10番のエージェントを起点とする4つの行動の連鎖を表している。本稿では、このようなサッカーにおけるボールを扱う行動の連鎖を扱う。フレームワークではエージェント間の意図の共有は想定されておらず、図中のすべての行動は10番のエージェントによって生成・評価されている。よって、10番のエージェントが想定した行動を他のエージェントが実行する保証は無い。

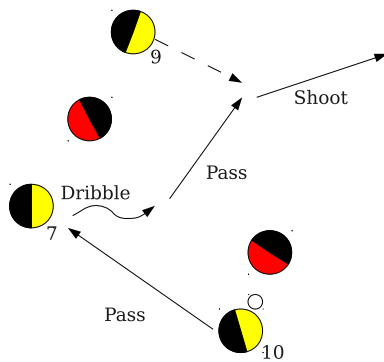


図1 複数のサッカーエージェントによる行動連鎖のイメージ図。10番から7番へのパス、7番によるドリブル、7番から9番へのパス、9番によるシュート、という4つの行動を連続して実行するプランを表す。

採用する探索木の模式図を図2に示す。図で示すように、フレームワークでは多分探索木によって探索処理を実行する機能が提供されている。図中のノードには、それぞれの ActionStatePair が格納される。木のルートノードから葉ノードまでのノード列をつなげると、ある行動連鎖が得られる。

探索木の走査アルゴリズムとして、単純な最良優先探索が使用される。ノードが生成されると、評価関数によってノードの評価値が得られる。この評価値は、最良優先探索のためのヒューリスティック値としても利用される。新規ノードが追加されるごとに、評価値に基づいてノード列を格納する優先順位付きキューが更新される。ノードの走査は優先順位付きキューでの格納順に実行される。実際に生成した行動連鎖をシミュレータの画面上に重ねて表示した様子を図3に、生成された行動連鎖の一部を評価値と共に列挙した様子を図4に示す。

本稿では、HELIOS Base [6] に実装されている行動連鎖探索フレームワークを用いる。HELIOS Base は RCSS で利用できるサンプルチームである。実行するチームとしても HELIOS Base をそのまま使用する。

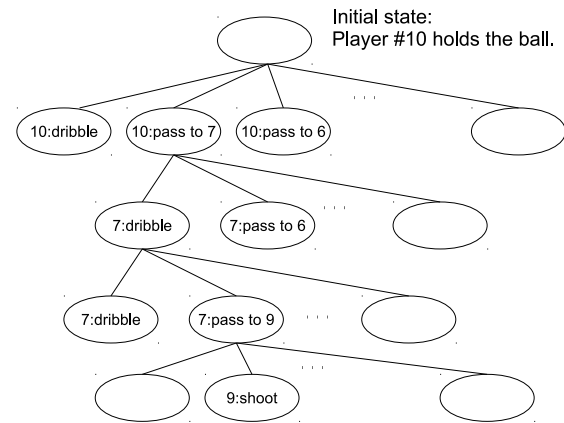


図2 フレームワークで採用する探索木の模式図。多分探索木を採用する。各ノードは行動とその結果の予測状態のペアを格納する。各ノードが持つ予測状態に基づいて取りうる行動が生成され、子のノードが作られる。

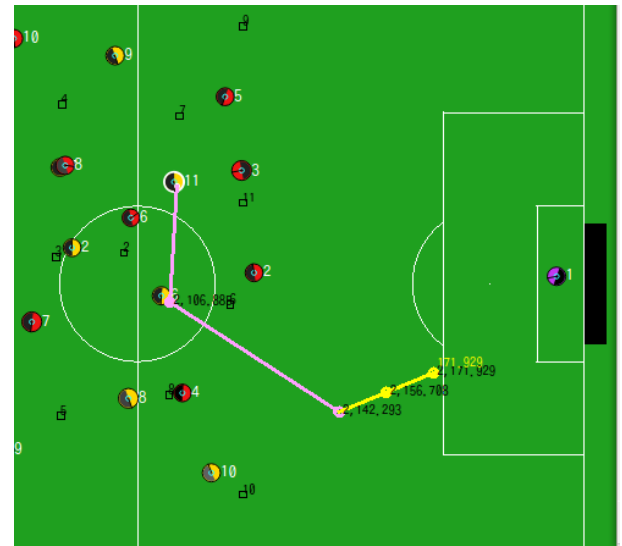


図3 実際に生成された行動連鎖の例。11番のエージェントにより、2回のパスと2回のドリブルがプランされている。

### 3.2 ランキング学習の適用

行動連鎖探索フレームワークでは、サッカーエージェントが取りうる行動を生成・列挙し、評価値がもっとも高い行動が実行される。列挙された行動はそれぞれが評価値を持っており、順位付けが可能である。この順位付けがチーム開発者の意図になるべく近いものとなるように、行動評価関数が設計される必要がある。この問題に対して、ランキング学習の適用が有望である。

ランキング学習は、情報検索システムを代表に様々な分野へ応用されている手法である [3], [11], [12]。情報検索システムと照らしあわせれば、クエリはサッカーエージェントが信念として持つフィールドの状態、検索結果は列挙された行動列の候補と捉えることができる。

本稿では、行動連鎖探索フレームワーク中で用いる評価関数を獲得するために SVM-light [2] を用いる。今回は以下の6つの特徴量を使用する。

| Unum=11 Time=[550, 0] 721 hits |         | Filter ID(s): | Length: | String(s):                                                                                       |
|--------------------------------|---------|---------------|---------|--------------------------------------------------------------------------------------------------|
| ID                             | Value   | L             | Seq     | Description                                                                                      |
| 50                             | 171.929 | 4             | 1       | [ 0: 1 pass (Lead[184]) t=6 from[11](4.64 -11.43)-to[6](3.83 2.15), safe=2 value=106.885260]     |
|                                |         |               | 41      | [ 1: 41 pass (Voronoi[1]) t=26 from[6](3.83 2.15)-to[10](23.74 15.07), safe=2 value=142.292583]  |
|                                |         |               | 48      | [ 2: 48 dribble (Simple[4]) t=35 from[10](23.74 15.07)-to[29.28 12.77], safe=2 value=156.708033] |
|                                |         |               | 50      | [ 3: 50 dribble (Simple[6]) t=44 from[10](29.28 12.77)-to[34.82 10.48], safe=2 value=171.929266] |
| 52                             | 171.081 | 4             | 1       | [ 0: 1 pass (Lead[184]) t=6 from[11](4.64 -11.43)-to[6](3.83 2.15), safe=2 value=106.885260]     |
|                                |         |               | 41      | [ 1: 41 pass (Voronoi[1]) t=26 from[6](3.83 2.15)-to[10](23.74 15.07), safe=2 value=142.292583]  |
|                                |         |               | 47      | [ 2: 47 dribble (Simple[3]) t=35 from[10](23.74 15.07)-to[27.98 10.83], safe=2 value=155.711767] |
|                                |         |               | 52      | [ 3: 52 dribble (Simple[8]) t=44 from[10](27.98 10.83)-to[33.52 8.53], safe=2 value=171.080809]  |
| 49                             | 171.081 | 4             | 1       | [ 0: 1 pass (Lead[184]) t=6 from[11](4.64 -11.43)-to[6](3.83 2.15), safe=2 value=106.885260]     |
|                                |         |               | 41      | [ 1: 41 pass (Voronoi[1]) t=26 from[6](3.83 2.15)-to[10](23.74 15.07), safe=2 value=142.292583]  |
|                                |         |               | 48      | [ 2: 48 dribble (Simple[4]) t=35 from[10](23.74 15.07)-to[29.28 12.77], safe=2 value=156.708033] |
|                                |         |               | 49      | [ 3: 49 dribble (Simple[5]) t=44 from[10](29.28 12.77)-to[33.52 8.53], safe=2 value=171.080809]  |
| 51                             | 169.392 | 4             | 1       | [ 0: 1 pass (Lead[184]) t=6 from[11](4.64 -11.43)-to[6](3.83 2.15), safe=2 value=106.885260]     |
|                                |         |               | 41      | [ 1: 41 pass (Voronoi[1]) t=26 from[6](3.83 2.15)-to[10](23.74 15.07), safe=2 value=142.292583]  |
|                                |         |               | 47      | [ 2: 47 dribble (Simple[3]) t=35 from[10](23.74 15.07)-to[27.98 10.83], safe=2 value=155.711767] |
|                                |         |               | 51      | [ 3: 51 dribble (Simple[7]) t=44 from[10](27.98 10.83)-to[32.22 6.58], safe=2 value=169.392393]  |
|                                |         |               | 2       | [ 0: 2 pass (Direct[3]) t=6 from[11](4.64 -11.43)-to[6](2.96 1.48), safe=1 value=101.014686]     |

Close

図 4 図 3 の状態で生成された行動連鎖の一部。value の列は各行動連鎖の評価値を表す。

|     |       |      |      |      |      |      |      |
|-----|-------|------|------|------|------|------|------|
| 74  | qid:1 | 1:73 | 2:25 | 3:32 | 4:23 | 5:49 | 6:50 |
| 82  | qid:1 | 1:62 | 2:9  | 3:40 | 4:50 | 5:49 | 6:50 |
| 79  | qid:1 | 1:67 | 2:16 | 3:36 | 4:37 | 5:49 | 6:50 |
| 77  | qid:1 | 1:67 | 2:16 | 3:36 | 4:62 | 5:49 | 6:50 |
| 100 | qid:2 | 1:51 | 2:16 | 3:53 | 4:26 | 5:49 | 6:50 |
| 99  | qid:2 | 1:54 | 2:19 | 3:51 | 4:21 | 5:49 | 6:50 |
| 86  | qid:2 | 1:64 | 2:12 | 3:38 | 4:41 | 5:49 | 6:50 |
| 82  | qid:2 | 1:62 | 2:9  | 3:40 | 4:50 | 5:49 | 6:50 |

図 5 訓練データの例。

- ・ 予測状態のボールと相手ゴールとの距離
- ・ 予測状態のボールとエージェント自身との距離
- ・ 予測状態のボール X 座標
- ・ 予測状態のボール Y 座標
- ・ 予測状態のエージェント自身の X 座標
- ・ 予測状態のエージェント自身の Y 座標

訓練データの例を図 5 に示す。各行は生成された行動列の候補に対する評価値と特徴量である。1 列目が評価値、2 列目がクエリ ID、3 列目以降は各特徴量を意味する。クエリ ID とは、SVM-light を情報検索に適用した場合にクエリごとに割り振られる ID である。よって、サッカーエージェントが持つフィールドの状態ごとに異なるクエリ ID が割り当てられる。すなわち、クエリ ID が異なることは、異なる状況下であることを意味する。

学習とエージェント実行の手順は以下ようになる：

- (1) 訓練データの作成
- (2) SVM-light により訓練データからモデルデータを生成
- (3) サッカーエージェント起動時にモデルデータ読み込み

現在の実装では、チーム中の全エージェントが共通のモデルデータを使用する。訓練データは人手で作成することを想定している。学習は試合実行前にバッチ処理で行われ、試合中のモデルデータ更新は行われない。

現状の手法では、訓練データの増加に対して学習にかかる時間が指数関数的に増大する問題がある。SVM-light では、訓練データの各行でペアを作り pair-wise 法で学習を行う。計算量は  $O(2^n)$  となるため、訓練データが 1000 程度でも現実的な時間で学習を終了させることはできない。このため、訓練データを蓄積して利用することが困難となっている。

#### 4. まとめ

本稿では、RoboCup サッカー 2D シミュレーション環境において選手エージェントの行動評価関数を獲得するために、ランキング学習を適用する手法を提案した。

今後の課題としては、継続的な学習と訓練データ作成の省力化が上げられる。現在の実装では訓練データのサイズを小さく抑えなければならないため、訓練データを蓄積して学習を行うことが困難となっている。また、訓練データ作成を省力化するためのツール開発も必要である。

#### 参考文献

- [1] Thomas Gabel, Martin Riedmiller and Florian Trost: A Case Study on Improving Defense Behavior in Soccer Simulation 2D: The NeuroHassle Approach. RoboCup 2008: Robot Soccer World Cup XII. pp. 61–72, (2008).
- [2] Thorsten Joachims: Making large-scale support vector machine learning practical, Advances in kernel methods, pp.169–184, MIT Press, (1999).
- [3] Thorsten Joachims: Optimizing Search Engines Using Clickthrough Data, Proc. of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '02, pp.133–142, (2002).
- [4] Hidehisa Akiyama, Tomoharu Nakashima, Shigeto Aramaki: Online Cooperative Behavior Planning using a Tree Search Method in the RoboCup Soccer Simulation,

- Proc. of 4th IEEE International Conference on Intelligent Networking and Collaborative Systems (INCoS-2012), (2012).
- [5] Hidehisa Akiyama, Itsuki Noda: Multi-Agent Positioning Mechanism in the Dynamic Environment, RoboCup 2007: Robot Soccer World Cup XI, pp.377-384, (2008).
  - [6] Hidehisa Akiyama, Tomoharu Nakashima: HELIOS Base: An Open Source Package for the RoboCup Soccer 2D Simulation, RoboCup 2013: Robot World Cup XVII, pp.528-535, (2014).
  - [7] Itsuki Noda, Hitoshi Matsubara: Soccer Server and Researches on Multi-Agent Systems, Proc. of IROS-96 Workshop on RoboCup, pp.1-7, (1996).
  - [8] Luis Paulo Reis, Nuno Lau, Eugenio Olivéira: Situation Based Strategic Positioning for Coordinating a Simulated RoboSoccer Team, Balancing Reactivity and Social Deliberation in MAS, pp.175-197, (2001).
  - [9] Peter Stone, Manuela Veloso: Task Decomposition, Dynamic Role Assignment, and Low-Bandwidth Communication for Real-Time Strategic Teamwork, Artificial Intelligence, (1999).
  - [10] Peter Stone, Richard S. Sutton, Gregory Kuhlmann: Reinforcement Learning for RoboCup-Soccer Keepaway, Adaptive Behavior, Vol.13, No.3, pp.165-188 (2005).
  - [11] 末廣 大貴, 畑埜 晃平, 坂内 英夫, 瀧本 英二, 竹田 正幸: SVM による 2 部ランキング学習を用いたコンピュータ将棋における評価関数の学習, 電子情報通信学会論文誌, J97-D(3), pp.593-600 (2014).
  - [12] 数原 良彦, 片岡 良治: 素性推定器を用いたランキング学習, 第 24 回人工知能学会全国大会予稿集, 2A1-04, (2010)