1B3-3

サッカーシミュレータ環境における模倣による行動の学習

Imitation Learning on a Soccer Simulator

高橋 昌生[*1] Masaki TAKAHASHI
小笠原 真吾[*1] Shingo OGASAWARA
山本 晃義[*1] Akiyoshi YAMAMOTO
野田 五十樹[*2] Itsuki NODA
岡 夏樹[*1] Natsuki OKA

[*1]京都工芸繊維大学 Kyoto Institute of Technology
[*2]産業技術総合研究所 National Institute of Advanced Industrial Science and Technology

In multi-agent systems, one of the difficulties of learning is to provide examples for training. In this study, we thus trained agents not by supervised learning, but by imitation learning from the logs of other teams. We used the RoboCup Soccer Simulator, and the aim was to acquire the pass play which is one of typical cooperative behaviors. We made experiments on the acquisition of 1) the ball position in dribble, 2) the judgement on whether the agent should dribble, and 3) the selection of the receiver of a pass. The experiments demonstrated the effectiveness of our learning technique.

1. はじめに

マルチエージェントシステムにおいて学習を行う場合，学習の問題の一つとして学習例の提示の難しさがある．教師付き学習を行う場合，正しい入力と出力のペアを与えてやる必要があるが，多数のエージェントに対して人間が十分な数の正しい入力と出力のペアを与えてやることは非常に困難である．そこで本研究では他の優秀なチームの行動のログを用いて，他チームの行動の模倣学習を行う．模倣学習とは，他のエージェントの行動の良い事例を集めて，それを模倣することにより能力を改善しようという学習手法のことである．

本研究では，ドリブルと，パスをするプレイヤによるパスを受けるプレイヤの選択という 2 種類の事柄について模倣による獲得を試みた．

2. ドリブルの学習

2.1 問題設定

ドリブルという行動は，ボールを蹴った直後からエージェントが得る状況が変わってしまう為，時系列を扱う学習となり学習モデルが複雑になってしまう。したがって，エージェントが

1. どのような状況でドリブル行動を開始するかを解析・学習
2. ドリブル中にボールを体に対してどの位置に置いているかを解析・学習
3. どの方向にドリブルするかを解析・学習
4. どのような状況でドリブル行動を終了するかを解析・学習

などのようにドリブル行動をいくつかの基本動作に分割して学習を行う必要がある。

また，ドリブルは，‘ ボールをある方向にドリブルで運ぶ ’という一連の動作であり，それはエージェントがボールと共にある方向へ移動するという状態を達成しようとする考え（意図）を表す。具体的には，そのエージェントが

- ボールを持っている
- あるスペースに向かおうとしている
- 何らかのサポートを必要としている（その方向に近い位置でパスを受けるプレイヤなど）

などの条件を満たしている状態（抽象化された状況）を表していると考えられる [1]。これらをログファイルからの解析に置き換えると

- ボールとエージェントの距離が 0 以上 *kickable margin* 以下である
- あるスペースに敵エージェントがいない
- あるスペースに味方エージェントがいる

などのように解釈できる。このような抽象化された状況の分類は，ログファイルからの解析の場合，そのほとんどが視覚センサからの情報に頼らざるを得ない。聴覚センサからの情報は，実際にエージェントがその情報を受け取ったかどうかを確認する術がなく，また say コマンドは (say ”)3G4c55¡c1”) などのように暗号化されており容易に解析できないからである。

2.2 ニューラルネットワークによる学習実験

今回は RoboCup 2004 の優勝チームである STEP04 のドリブルを模倣する。このチームはドリブルで強引に中央突破を図り，ゴール前に走り込んできた味方にパスしてシュートを決めるというパターンでゴールを量産するチームで，特にそのドリブルには特徴がある。STEP04 の RoboCup2005 のログファイルから必要な情報を抽出し, 2.1 節で述べたドリブルの基本動作をニューラルネットワークを使用して学習させる。

実験を始める前に STEP04 の RoboCup2005 全 14 試合のログファイルからドリブルをしている状況とドリブルをしていない状況での $dist < unum\ too\ far\ length$ (RoboCup Soccer Simulator においてプレイヤのチーム名が常に見える範囲) の視界の範囲内における平均プレイヤ数を比較してみた。ドリブル区間の抽出には Team Assistant 2003 [2]（以下 TA2003）のログ解析機能を用いた。表 1 より，ドリブルをしている状況の方がドリブルをしていない状況よりも味方・敵プレイヤ共に少なくなっていることがわかる。これは視界の範囲内のプレイヤ数がドリブルという行動の動機付けとして関与していることを示唆している。

2.3 実験 1-1

まず，STEP04 のプレイヤがドリブル中にボールを体に対してどの位置に置いているか解析し，ニューラルネットシミュ

連絡先: 山本 晃義, 京都工芸繊維大学 大学院工芸科学研究科, e-mail: yamamoto@vox.dj.kit.ac.jp

	ドリブル	ドリブル外
味方プレイヤ数	0.617	3.343
敵プレイヤ数	1.798	4.720

表 1: $dist < unum\ too\ far\ length$ の視界の範囲内における平均プレイヤ数

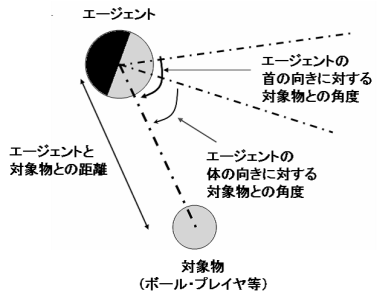


図 1: エージェントと対象物の距離と角度

レータ tlearn[3] を用いて学習を行った。ドリブルの場合，ドリブルを行うエージェントから見て比較的近距離にいるプレイヤの分布が重要になってくる。したがって，ニューラルネットワークの入力層は，ドリブルするエージェントから見て $dist \leq unum\ far\ length$ (RoboCup Soccer Simulator においてプレイヤのチーム名と背番号が常に見える範囲) の

- 左側に存在する味方プレイヤの数
- 右側に存在する味方プレイヤの数
- 左側に存在する敵プレイヤの数
- 右側に存在する敵プレイヤの数

の 4 ノード，出力層は

- ドリブルするエージェントの体に対するボールの角度（ラジアン表記）(図 1)

の 1 ノード，中間層は 2 ノードとした (図 2)。RoboCup2005 の STEP04 の全 14 試合のログファイルのうち 9 試合を学習に使用し，残り 5 試合はテスト用データとして残しておく。観測者エージェントはチーム内で一番ドリブル数の多い背番号 10 のプレイヤである。教師信号は 436 例，学習率 0.1 で 10000 回の学習を行った。

2.4 実験 1-2

次に，ドリブルをする状況とドリブルをしない状況の学習を行った。ニューラルネットワークの入力層は，ドリブルを行うエージェントから見て $dist \leq unum\ far\ length$ の

- 左側に存在する味方プレイヤの数
- 右側に存在する味方プレイヤの数
- 左側に存在する敵プレイヤの数
- 右側に存在する敵プレイヤの数
- ボールが $kickable\ margin$ 以下に存在するか
  （存在する：0，存在しない：1）

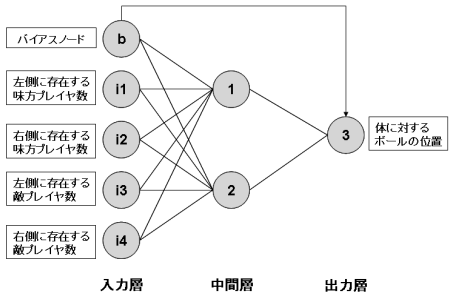


図 2: 学習モデル（実験 2-1）

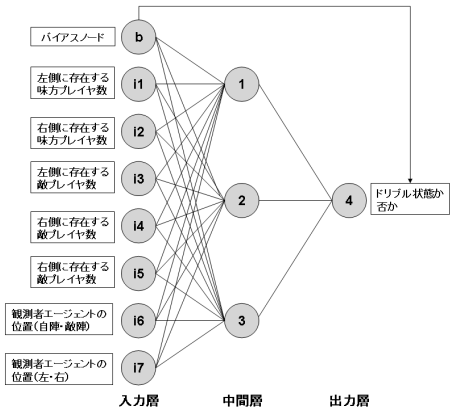


図 3: 学習モデル（実験 2-2）

の 5 ノードと

- 観測者エージェントプレイヤの位置（自陣：0，敵陣：1）
- 観測者エージェントの位置（攻撃方向に向かって左側：0，右側：1）

の 2 ノード（計 7 ノード），出力層は

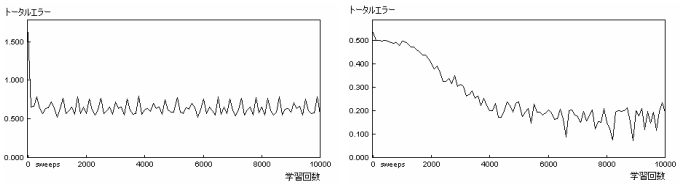
- ドリブルしているかしていないか
  （ドリブルをしていない：0，ドリブルをしている：1）

の 1 ノード，中間層は 3 ノードとした (図 3)。実験 1 同様，RoboCup2005 の STEP04 の全 14 試合のログファイルのうち 9 試合を学習に使用し，残り 5 試合はテスト用データとして残しておく。観測者エージェントはチーム内で一番ドリブル数の多い背番号 10 のプレイヤである。教師信号は 996 例（ドリブル状態は 436 例，ドリブル以外の状態は 560 例），学習率 0.1 で 10000 回の学習を行った。

この実験ではログファイルのデータをそのまま使用するとドリブル状態は 436 例，ドリブル以外の状態は 55575 例であり，ネットワークがこのようなデータに対して平均二乗誤差を小さくしようとすると，すべてドリブル以外の状態であると答えれば良くなることは明白である為，ドリブル以外の状態をランダムに抽出し訓練例の数をある程度揃えるようにした。

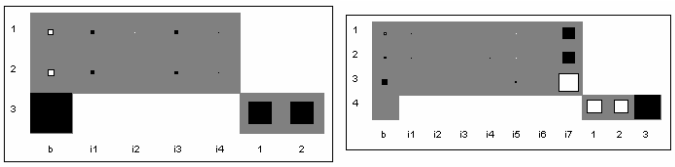
2.5 実験結果

実験 1-1，実験 1-2 の結果をそれぞれ図 4(a)，図 4(b) に示した。これらは出力ノードの平均二乗誤差の推移を示す。



(a)実験1-1　(a)実験1-2

図 4: 学習曲線



(a)実験1-1　(a)実験1-2

図 5: 結合加重

さらに各々の結合加重の様子を図 5 に示す。これらは横軸ノードから縦軸ノードへの結合の様を表している。結合がある場合は背景色が灰色に，結合がない場合は背景が白色になる。黒の四角はマイナスの結合を，白の四角はプラスの結合を表し，四角の大きさは結合強度の絶対値を表している。’b ’はバイアスノードである。結合加重はノード間の作用の強さを表し，どれだけ関係が深いかが視覚的に理解できる。

2.6 実験結果の考察

実験 1-1，実験 1-2 とも，学習曲線を見ると，エラーはまだ残っているが、ある程度学習が進んだ様子が分かる (実験 1-1 では学習開始直後にエラーが減少している)。ただし，結合加重を調べてみると，必ずしも，合理的な判断で行動が決まっているとは言えない面もあり，今後さらに，入力情報や学習データの適切さ，入力と出力の相関関係の分析，模倣対象チームの動作アルゴリズムの分析等を行う予定である。

3. パスの送り先の学習

3.1 問題設定

この実験で模倣する行動は，ボールを持ったプレイヤが他のプレイヤに対してパスを送る場合，どのプレイヤに対してパスを送るかの選択である。選択は，ボールを持ったプレイヤとの距離により何番目に近いプレイヤに対してパスを送ったかという選択である。なお，6 番目以降の距離のプレイヤにパスを送るプレイはその他として扱った。模倣の対象となるチームは，UvaTrilearn である.

3.2 使用した特徴量

この実験で使用した特徴量は，ボールを持ったプレイヤからの距離が近いプレイヤの行動の意図を手作業により抽出したもので，最も近いプレイヤから 5 番目に近いプレイヤまでの 5 つを，距離の近いプレイヤから順に状態 1，状態 2，状態 3，状態 4，状態 5 として使用した。

5 番目に近いプレイヤまでの情報を使用したのは，本研究で使用したデータにおいて 5 番目に近いプレイヤまでにパスを送っている割合が約 95.9%を占めるためであり，これ以上遠いプレイヤの情報は行動選択には大きな影響を及ぼさないのではないかと考えたためである。

3.3 使用したデータ

Soccer Simulator を使用して実際に試合を行い，そのログを学習データとして使用した。使用したデータは 3 試合分であり，試合 1，試合 2，試合 3 から抽出したパスの総数はそれぞれ 98，95，101 である。各試合に含まれていたパスの種類 (何番目に近いプレイヤにパスを出していたか) を表 2 に示す。

表 2: 各試合に含まれるパスの種類

	1 番目	2 番目	3 番目	4 番目	5 番目	その他
試合 1	43	24	14	4	9	4
試合 2	27	36	14	8	7	3
試合 3	34	42	13	4	3	5
合計	104	102	41	16	19	12

3.4 行動意図の単位

この実験ではボールを持たないエージェントの行動意図を足元にパスをもらう，バックパスをもらう，ポストプレー，サイド突破，ゴールを狙う，ゴール前に移動，ポジションに戻る，ポジションをキープ，と設定した。

3.5 実験 2-1

試合のログファイルよりパスのタイミングを抽出し，そのタイミングにおいてボールを持ったプレイヤがどのプレイヤにパスを出すか，という選択を学習し，その精度を調べた。実験データとして 3 試合分のログファイルを使用し，学習に用いるデータ数を増やしながら精度の推移を調べた。学習アルゴリズムは AODE[4] を使用し，学習にはデータマイニングツール Weka[5] を用いた。評価には 10 fold のクロスバリデーションを用いた。

この実験では，意図が適切にラベル付けされている場合にそれを特徴量として正しく学習することが出来るかどうかを確かめるために，学習データの他のエージェントの意図は試合の様子をログプレイヤを用いて再生しながら設計者がラベル付けしたデータを用いた。

3.6 実験 2-2

次に，実験 2-1 では手動で抽出していた意図を自動的に識別するための意図識別器を作成し，それを用いて学習を行った．意図識別器は図 6 のアルゴリズムで実装し，手動での識別に近くなるよう以下の学習手順で閾値を調整した．こうして作成した意図識別器によって抽出した意図と手動で抽出した意図との一致率は 60.6% となった.

1. 閾値を 0 から 0.1 刻みに増加させていき，それぞれの値において 3 試合分のデータの意図推定を行う.
2. 3 試合分の識別結果と手動で行った意図のラベル付けとの一致率が最も高いものを閾値として選択する．一致率が最も高い閾値が複数存在する場合，その中で中間にあるものを選択する.
3. 次の閾値に対してこの手順を繰り返す.
4. 閾値が安定するまで以上の手順を繰り返す.

意図を識別したい味方のプレイヤをターゲットとして:

1. もしターゲットのプレイヤとボールを持ったプレイヤとの間の線分から敵プレイヤの距離が a 以下なら “ポジションをキープ” を出力する。
2. もしターゲットのプレイヤとオフサイドラインの距離が b 以下の場合
   (a) もしターゲットのプレイヤの y 座標が-c より上，c より下の場合 “ポストプレイ” を出力する。
   (b) それ以外の場合，“サイド突破” を出力する。
3. もし $x > d$ かつ $-e < y < e$ の場合，“ゴールを狙う” を出力する。
4. もしボールを持ったプレイヤよりも f 以上後ろなら “バックパスをもらう” を出力する。
5. それ以外の場合，“足元にパスをもらう” を出力する。

図 6: 意図推定のアルゴリズム (a から f は閾値であり，学習により値を調整した)

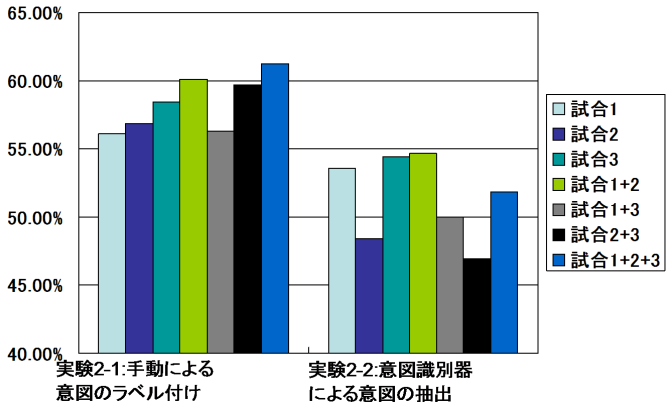


図 7: 実験 2-1 と実験 2-2 の結果

3.7 結果

3.5 節，3.6 節に従い，実験を行った。この実験における正答率とは，クロスバリデーションによって未知の学習データを与えた場合にどれだけ正確な行動を出力することができたか，である。

実験結果を図 7 に示す。

3.8 実験 2-1 の考察

図 7 より，データ数を増加させることにより徐々に正答率が上昇している傾向にある。しかし，本実験では事例数が 295 と少なく，事例数が不足しているため，精度を上げるためにはさらにデータを増やす必要がある。

次に，全試合分のデータを用いた学習結果において，行動ごとに正答率を算出してみた。結果を表 3 に示す。この表より，事例数の多い 1 番近いプレイヤへのパスや 2 番目に近いプレイヤへのパスは比較的高い精度で選択出来ているが，事例数が少ないプレイになるにしたがって正答率が下がっている。今後事例数を増加させ，これらのプレイの選択の精度も向上させたい。

また，実験データに用いた事例の中で状態が同じものを集めてみた。このとき，同じ状態であっても取っている行動が異なる事例が少なからず見られた。これを改善するためには，意図の分類を見直す必要があるが，あまり細分化しすぎると今度は学習に必要な事例数が増加してしまう。

また，機械的に認識しにくい意図を採用してしまうと自動的に意図を判別することが難しくなってしまう。適切な意図の分類の設計をどのようにして行うのか，という点について検討を行う必要がある。

表 3: 行動別の正答率

	事例数	正答数	正答率
1 番目	104	76	73.1%
2 番目	102	72	70.6%
3 番目	41	22	53.7%
4 番目	16	5	31.3%
5 番目	19	5	26.3%
その他	12	0	0.00%

3.9 実験 2-2 の考察

図 7 より，実験 2-1 に比べて実験 2-2 の方が全体的に精度が低い傾向にある．これは，意図識別器の性能がそれほど良くないために，手動で意図のラベル付けをしたデータに比べて状況の抽象化が適切に行われていないせいであると考える．

実際の環境においてはエージェントに対して他のエージェントの意図は与えられず，センサ入力から意図を識別する必要があるため，意図の自動識別の精度を向上させることは非常に重要な課題の一つであり，今後更に意図の識別率を向上させる必要がある．

また，今回は意図識別器の閾値を設定する際に精度の評価基準として手動による意図の抽出結果を使用したが，それは必ずしも適切なものではなく，より適切な手法を用いて意図識別器の閾値を設定して実験を行う必要がある．

参考文献

[1] 野田五十樹：HMM による協調動作の模倣学習，pages 3D1-04， 人工知能学会全国大会（第 17 回）論文集, 人工知能学会，2003

[2] Team Assistant (TA) 2003: `http://www.sbcee.net/pres/index.htm`

[3] Tlearn software:`http://crl.ucsd.edu/innate/tlearn.html`

[4] Geoffrey I. Webb and Janice R. Boughton and Zhihai Wang: Not So Naive Bayes: Aggregating One-Dependence Estimators., Machine Learning, 58(1), pp. 25–32 (2005)

[5] Weka 3 - Data Mining with Open Source Machine Learning Software in Java: http://www.cs.waikato.ac.nz/ml/weka/