協調行動の学習における合意の安定化手法

太田正幸
東京工業大学 情報理工学研究科 数理・計算科学専攻
m-oota@is.titech.ac.jp

Abstract

Because of the complexity of multi-agent systems, machine learning techniques are indispensable. When the agents cooperate with each other, they have to make agreement about next behaviors. But, if some good cooperative strategies have almost the same effectiveness, agreement could be unstable, and efficiency of learning will reduce.

In this paper, we show that agent can avoid this problem with incremental learning. We evaluate the effectiveness using a case study of learning pass play on the Soccer Server.

1 はじめに

近年、複数のエージェントが協調的に動作することにより、単一のエージェントでは達成することの出来ない仕事を行ったり、組織的に行動することで効率を高めるといったプログラミング手法の重要性が認識されつつある[5][6]。しかし、人間が各問題に対応した協調行動を完全に記述することは困難であることが多く、手間のかかる作業でもあるため、動作の詳細は機械学習により獲得することが望まれる。

複数のエージェントが協調行動を学習する場合、ある状況に対し同程度に効果的な複数の戦略が存在することがあるため、協調する各エージェント間での合意が不安定になり、学習の効率が下がることがある。

本稿では、マルチエージェント環境において強化学習を行う際、各エージェントが段階的に学習結果を適用しながら学習を行うことでこの問題を解決することが出来ることを示し、近年、マルチエージェント環境における学習を扱うための標準問題となりつつある、コンピュータ上でサッカーの試合をシミュレートする例[2]をもとに、この方法が有効であることを検証する。

2 協調行動を学習する際の問題点

2.1 対象とする問題

複数のエージェントが、次のような環境において協調行動を学習することを考える。

- エージェントどうしが直接通信することは出来ない。
- 各エージェントは、グローバルな視点からの情報を持たずに、独自の視点から得られる情報だけをもとに行動する。
- 各エージェントの知覚情報にはノイズが入る。

コンピュータ上でサッカーの試合をシミュレートする例題は、マルチエージェント環境における標準問題の1つになりつつあるが、中でもSoccerServer[4]は、上記のような環境を扱いやすい形で実現している。本研究では、SoccerServerを用いて、図1に示すように、攻撃側と防御側それぞれ2人ずつに分かれ、攻

撃側のエージェント A_1 が防御側のエージェント O_1, O_2をかわしてもう片方の攻撃側のエージェント A_2にパスを出し、A_2がゴールにシュートする」というサッカーの練習の 1 場面をシミュレートする例をもとに、攻撃側の 2 つのエージェントに協調行動を学習させるという例題を用いる。この状況には、

- S_1(スルーパス)：
  A_1が O_1, O_2の間を通してパスを出し、同時に A_2は点 P まで走ってボールを受け取り、シュートする。
- S_2(壁パス)：
  A_1が A_2の現在地に向けてパスを出し、A_2はその場でパスを受け取り、シュートする。

という 2 つの同程度に効果的な戦略があり、A_1, A_2が独自の視点からの敵情報をもとに、どちらの戦略を選ぶべきかを同時に学習させる際の問題を扱う。

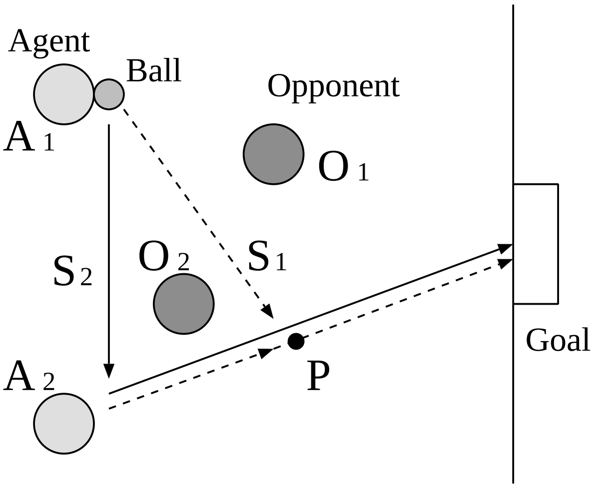

図 1: 問題として扱う場面

2.2 複数のエージェントが同時に学習を行なう場合の留意点

ある環境において、単一のエージェントが学習を行なう場合と、複数のエージェントが同時に学習を行なう場合とでは、次のような違いがある。

- 単一のエージェントが学習を行なう場合
  ある状況に対し、2 つの戦略が有効であると学習したとき、同じ状況であれば 2 つの戦略はどちらも成功する可能性が高い。従って、どちらの戦略を選択しても戦略の成功率は変わらない。また、敵も同時に学習する場合を想定して、意図的にその 2 つの戦略をランダムに選択することもある。
- 複数のエージェントが同時に学習を行なう場合
  図 1のような状況において、A_1は、上記と同様に、S_1, S_2という 2 つの戦略が有効であると知っており、同時に A_2も S_1, S_2という 2 つの戦略が有効であることを学習済みであるとする。このとき、どちらの戦略も過去に有効だったからといって、A_1が S_2、A_2が S_1を選択してしまうと、ボールは敵 O_2に取られてしまい、戦略は失敗する。これは、今までに成功した作戦であっても、次に、協調す

る相手が取る作戦と同じ作戦を取らない限り、失敗する可能性が高いことを意味する。また、学習の最中にも、協調する相手の行動次第で、正しい戦略を選んでいたとしても協調行動は失敗に終わることがあることを示している。当然、敵が同時に学習をする場合を想定して、戦略の選択にランダムな要素を入れると、協調の効率は著しく低下する。

従って、複数のエージェントが同時に学習を行う場合には、以下のような点に留意する必要がある。

- ある状況に対して、協調行動をする全てのエージェントが同一の戦略をとる（合意を形成する）ように、学習をしなければ成らない。
- ある状況に対して取った行動が失敗しても、戦略の選択を誤ったかどうかはわからないため、ランダムに行動してみて、うまく行ったときに報酬を与える強化学習が望ましい。
- エージェント間で直接通信が出来るような場合を除き、敵が同時に学習をする場合などを想定して戦略の選択にランダムな要素を入れない。

2.3 合意形成の際の問題点

図 1の状況で、A_1, A_2がランダムな戦略を実行し、うまく行ったときの知覚情報に対する戦略を教師信号として学習を進めると、図 2に示すように、同じ状況に対して、異なる 2 つの戦略を学習してしまうことがある。前述の通り、複数のエージェントが同時に学習を行う場合にはそのエージェント間に同一の戦略を取るための合意がなされている必要がある。しかし、エージェントどうしが直接通信することが出来ない環境では、過去の学習結果から合意を導き出すことになる。一般的には各状況に対して成功した学習回数が多い戦略を選ぶが、この回数に偏りが少ない状況に対しては、学習を終えた段階で各エージェントがどちらの戦略を取るようになっているか分からず、合意が安定しないため、協調行動の成功率が低下するという問題が生じることがある。

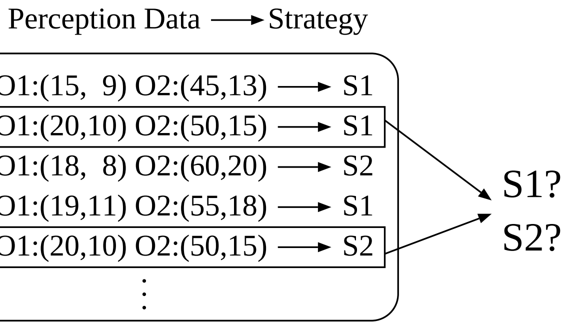

図 2: 合意形成を妨げる学習例

特に、ある状況に対し、S_1, S_2が同程度に効果的な戦略である場合には、このように異なる 2 つの戦略は長い間学習を続けても、学習回数にあまり偏りが生じないため、この状況に対して A_1, A_2が選ぶ戦略が定まらず、同一の戦略を取るように学習することは難しい。さらに、A_1, A_2がそれぞれ異なる独自の視点からの情報を元に、戦略を選ぶ場合には、A_1, A_2の状態空間の切り分けが異なり、合意形成が更に困難になる。

3　段階的な学習結果の適用

上記のような問題が生じるのは、学習を続けてもこの 2 つの戦略間に学習回数の偏りが生じないためである。従って、学習が進むにつれて成功した学習例の数が偏るように、意図的に学習例を作り出してやればよい。

ところで、学習回数の偏りがあまり生じないような状況においても、学習の初期段階では比較的学習例に大きな偏りが生じやすい。例えば、ある状況に対し、S_1を 2 回 S_2を 1 回学習した段階では、A_1, A_2は S_1に合意することが比較的容易である。従って、学習の初期に生じた学習例の偏りを増幅するように学習を続ければ、ある状況に対し、協調行動をする各エージェントが、実際にうまく行った学習例に偏った同一の戦略を最終的には取るように学習させることが可能である。

この考えに基づき、ランダムに行動して学習例を蓄えていくだけでなく、エージェント自身の判断を混ぜ、段階的に学習結果を適用しながら再学習を行うと、今までにうまく行った学習例を再び学習する回数が相対的に増え、学習例の偏りを増幅することになるため、先に述べたマルチエージェント環境特有の問題を解決することが出来ると考えられる。

4　実験

図 1で示した状況において、A_1, A_2それぞれが、独自の視点からの情報をもとに、協調行動を学習させる実験を次の手順で行なった。

- プレイヤーとボールを特定の位置に配置する。
- 各エージェントは自分の取る戦略を決定し、その戦略に沿って 3 秒間だけシミュレーションを行なう。
- 3 秒以内にシュートが出来た（シュートの関数を実行した）場合、協調が成功したとみなして、最初の状況に対して選んだ戦略を学習する。

以上を、敵の位置を毎回変化させながら繰り返し行なった。そのときの状況を図 3に示す。

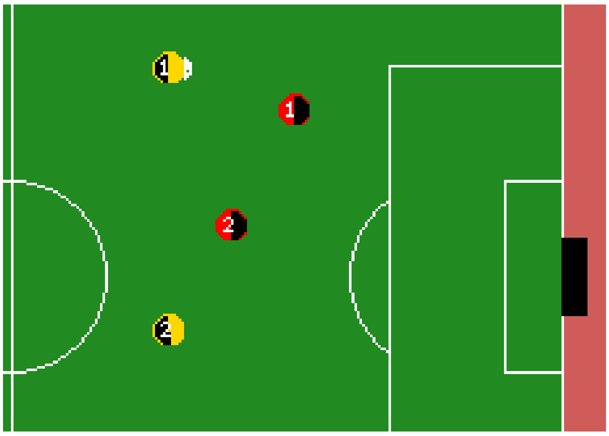

図 3: 実験の場面

各エージェント A_1, A_2は、隣どうしの Layer に含まれるノードが全て密に結合している 4 層の Neural Network を個別に持っており、同時に学習を行なわせた。この Neural Network は、敵 O_1, O_2の相対的な

方向と距離という４つの値を入力とし、戦略 S_1, S_2 のどちらを選択するべきかを決定するために０から１までの実数を１つ出力する。そして、S_1 でうまく行った場合には１、S_2 でうまく行った場合には０という教師信号を与え、back propagation により学習を行った。また、各中間層のノード数はいずれも３０個である。

防御側のエージェント O_1, O_2 には、RoboCup[2] の準優勝チームである Andhill[1] を用いた。

この学習を以下の３通りの方法で行い、比較した。

- A_1, A_2 とも、S_1, S_2 をそれぞれ 50% ずつの確率で行い、３秒以内に A_2 がシュートを行うことができた（シュートが入った時という分けではない）場合に、そのときの入力と戦略の組みを保存しておく。この作業を 2500 回繰り返し、後で Neural Network に 20000 回 back propagation で繰り返し学習させる。この作業を５回繰り返す。
- A_1, A_2 がそれぞれ 50%の確率で、自分の Neural Network からの出力で行動し、それ以外の場合は１つ目の方法と同様、ランダムに行動する。この方法でデータを取り、上記と同様の回数だけ学習を行う。
- ２つ目の方法と同様だが、自分の Neural Network からの出力の通りに行動する確率を、20000 回の back propagation 毎に、0%,20%,40%,60%,80%と増やしていく。

敵 O_1, O_2 の位置をそれぞれ２５通りずつに変化させた、合計６２５通りの状況について実験した結果、A_2 がシュートを行うことが出来た確率が表 1、ならびに図 4のように上昇し、最終的に、ランダムに学習を進めていっただけの場合に対して 14.7%程度効率が上がることが確認された。

なお、この実験では学習データのうち約 9.64%の状況に対し、２通りの戦略が学習されていた。

表 1: 実験結果

学習の回数 (回)	0	20000	40000	60000	80000	100000
ランダムに学習 (%)	23.4	40.8	49.0	51.2	50.8	51.6
50% 学習結果を適用 (%)	23.4	38.6	52.2	56.4	55.8	55.4
段階的に学習結果を適用 (%)	23.4	40.8	48.6	53.0	58.0	59.2

5　おわりに

ある状況に対し、複数の戦略の中から取るべき協調戦略を強化学習させる場合、成功した学習例の回数に大きな偏りがあれば、複数のエージェントが協調動作を学習することができる。しかし、同程度に効果的な戦略が存在し、成功した学習例の回数に偏りが少ないと、学習方法によっては複数のエージェント間の合意が不安定になり、学習の効率が下がることがある。

この問題を解決するためには、学習の成功例の偏りを増幅するように学習を進める機能をエージェントに持たせればよく、段階的に学習結果を適用しながら学習を行うことで実現することが出来る。

Soccer Server 上で以上のことを確かめるための実験を行った結果、段階的に学習結果を適用しながら学習を行うと、ランダムに学習を続けていった場合に比べて 14.7%程度の効率化が見られた。

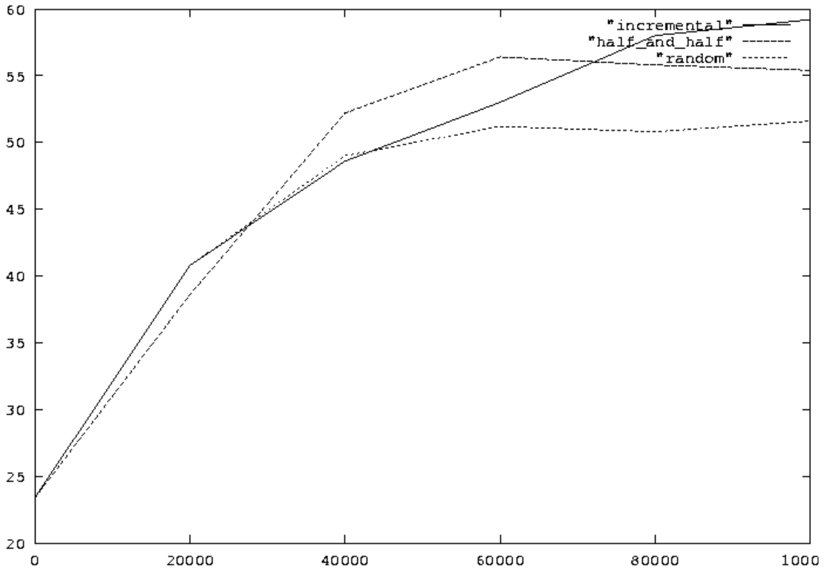

図 4: 実験結果

参考文献

[1] Tomohito Andou. “Refinement of Soccer Agents’ Positions Using Reinforcement Learning” proceedings of RoboCup-97 (to appear)

[2] H.Kitano, M.Asada, Y.Kuniyoshi, I.Noda and E.Osawa “Robocup: The robot world cup initiative” IJCAI-95 Workshop on Entertainment and AI/Alife pages 19-24 August 1995.

[3] Hitoshi Matsubara, Itsuki Noda, and Kazuo Hiraki. “Learning of cooperative actions in multiagent systems: a case study of pass play in soccer” AAAI. Adaptation, Coevolution and Learning in Multiagent Systems: Papers from the 1996 AAAI Spring Symposium, AAAI Technical Report SS-96-01, pages 63-67. Menlo Park, CA, March 1996, AAAI Press.

[4] Itsuki Noda. “Soccer server: a simulator of robocup” In Proceedings of AI symposium ’95, pages 29-34 Japanese Society for Artificial Intelligence December 1995.

[5] Peter Stone and Manuela Veloso. “Multiagent systems: A survey from a machine learning perspective” IEEE Transactions on Knowledge and Data Engineering, June 1996.

[6] Ming Tan “Multi-Agent Reinforcement Learning: Independent Vs. Cooperative Agents” Proceedings of the Tenth International Conference on Machine Learning pages 330-337 June 1993.