Learning Cooperative Behaviors in RoboCup Agents

Masayuki Ohta

Dept. of Mathematical and Computing Sciences,
Graduate School of Information Science and Engineering
Tokyo Institute of Technology
m-oota@is.titech.ac.jp

Abstract. In the RoboCup environment, it is difficult to learn cooperative behaviors, because it includes both real-world problems and multi-agent problems. In this paper, we describe the concept and the architecture of our team at the RoboCup'97, and discuss how to make this agent learn cooperative behaviors in the RoboCup environment. We test the effectiveness using a case study of learning pass play in soccer.

1 Introduction

Recently, multi-agent systems have become a large field of artificial intelligence.[3] Because of the complexity of multi-agent systems, machine learning techniques are indispensable.

Simulation of soccer game has become one of a standard problem of multi-agent systems, and the first RoboCup[1] was held. The official simulator for the RoboCup, Soccer Server[2], presents the following problems.

- **Real-world problems**
  - **Noisy environment**
    Perceptional information includes some errors. And action of agent is unreliable.
  - **Limited field of vision**
    Each agent can see only 90 degrees angle of forward. Then it have to estimate the backward situation
  - **Real-time processing**
    Because the ball will move, agents can not capture the ball at the current ball position. And, opponents obstruct our plan, one after another.
- **Multi-agent problems**
  - **Cooperative behaviors are advantageous**
    Some cooperative behaviors are very profitable[4]. The most obvious example here is pass play.
  - **No shared memory**
    When the agents cooperate, they have to make consensus of the way of cooperation. But in this case an agent have to guess another agent's action to cooperate.

- **No global view exist**
  Each agent have to decide the next action only from the perceptional information from their own standpoint. This make it difficult to behave cooperatively.

We aim at acquiring effective cooperative behaviors with machine learning in such environment. In such multi-agent environment, forming agreement is necessary. Regarding the real-time processing as important in this case, we inquire into the model that make agreement without direct negotiation.

2 Team Description

In this section, we describe our team which entered the RoboCup'97. We show noteworthy points to create soccer agents in the RoboCup environment, explain the relation to our agents' architecture, and discuss how can we improve our team based on the result of the RoboCup'97.

2.1 Design Issue

As mentioned above, the RoboCup environment includes a lot of features, and all of them are obstacle when we create soccer agent. To build strong soccer team, we programmed our agents paying attention to the following two points.

Cooperation using perceptional data from different stand point

There are a lot of approaches to make agreement with negotiation among the agents. But in this case, as we mentioned before, real-time processing is important. (Otherwise, opponents will steal the ball while our agents negotiate about next strategy.) Therefore our agents do not negotiate each other, and each agent decide next action only with information from their standpoint to succeed in the cooperative work.

Main problem here is the following two.

- An agent occasionally differ with others in assessment of the situation, because of observing from different standpoint,
- Agent's field of vision is restricted, and each agent gets limited visual information.

Learning cooperative behaviors

It is very difficult to describe agents' cooperative behavior accurately. Especially, in the RoboCup environment, there are no global view and forming agreement is going far difficult. So, it is necessary to get the way to behave cooperatively with machine learning technique. But in multi-agent environment, following new problems will occur, which do not take place in single-agent environment.

- All cooperating agents have to choose a correct strategy. If one of these agents choose incorrect strategy, the collaboration will fail. Then, some agents, which choose correct strategy, will learn that it is bad selection in this case, even if it is good.
- All cooperating agents have to choose the same strategy. Even if each cooperating agent chose a correct strategy, the collaboration will fail again, unless all agents have chose the same strategy. This problem could occur when there are more than two correct strategies.
- Each agent should learn only one strategy for a situation. If some good strategies have almost the same effectiveness, an agent can not fix the strategy for the situation. Then it is hard for cooperating agents to choose the same strategy, and causes reduction of efficiency.

2.2 Action Flow

Here, we show the details of our agents' action, and mention the relation to the issues previously stated.

Our agents' action is based on the rules of perception-action pair. When an agent receives visual information, it acts as the following flow.

- **Convert visual information into internal data structure**

- **If the ball is in the kickable range, kick it to the best way**
  - If it is near the opponent's goal, shoot the ball to the best way between the left goal post and the right goal post.
  - Else if there are some teammate whom the agent can pass the ball in safety, pass the ball to the farthest teammate.
  - Else kick the ball to the best way between the opponent's left corner and the right corner.

  The function to calculate the best way have three argument, left, right and distance. This function outputs the most open direction between left side and right side in the distance (Figure 1).
  Considering the real-time processing, our agents decide the best way only with the information already have. Then it occasionally pass the ball to opponents, because of lacking blind spot data. But sometimes succeed in a quick pass and shoot.
  We are planning to implement this module with machine learning.

- **Else if the agent have some special strategy, do the strategy**
  Special strategy is action, such as run to the open space in order to receive the pass. This module, also should be implemented by machine learning, but this strategy can succeed only when the cooperation with another agent which will pass the ball (using machine learning as previously stated) succeed. Then, the problems of learning cooperative behaviors will occur.

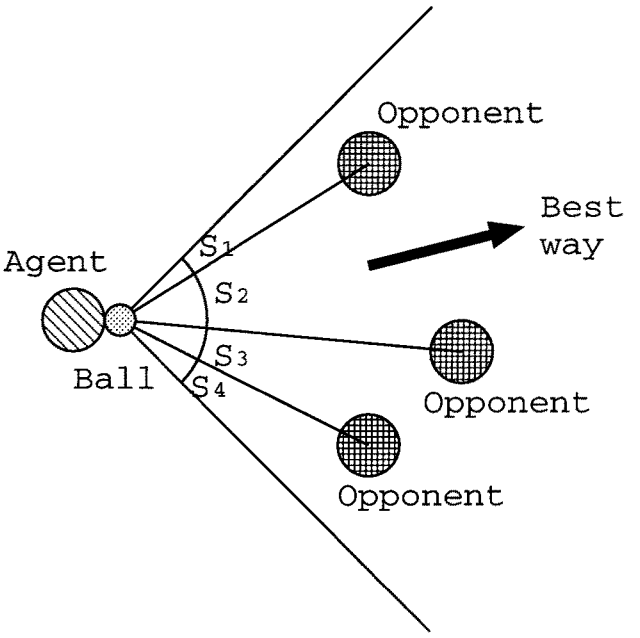

Fig. 1. Best way

- **Else if the agent is closest or second, chase the ball**
  Two agents chase the ball, in case one miss the ball. Our agents estimate future location of the ball considering current location and speed of it. Then it goes to the estimate position and capture the ball.
  If it does not use the estimate position, it lose the ball so often.

- **Else, trace the ball a while, then go back to their home position**
  In the RoboCup environment, agents sometimes lose the ball, because perceptional information is not reliable. Not to go back their home position immediately in case an agent lose track of the ball, it trace the ball a while.

2.3 Looking back over the RoboCup'97

Simulator league of the RoboCup'97 had 29 various types of teams, but all stronger teams had following features.

- They are very good at capturing the ball
- They move very fast

Our team was defeated by AT-Humboldt-97(World Champion team) in quarter final by a score of 14 to 7. In this match, we were beaten in the following pattern, though our agents also move quickly.

1. Can not find a teammate whom the agent can pass the ball safety.
2. Kick the ball to the most safety direction and far from our goal.
3. The ball is intercepted, when an opponent reached there earlier.

In this situation, if an agent can expect the position to where the teammate will kick the ball, it can go there earlier and could get the ball. Therefore, to make strong a team, besides the agent which have the ball learn where to kick it, other agents should learn the place where the ball will be kicked to.

3 Experiments

Learning cooperative behaviors in the RoboCup environment have many problems as we mentioned above. In this section, we show our approaches to the problems using a case study of making consensus about pass courses.

3.1 Learning of the pass play

We carried out the following situation.

- Two agents (A_1, A_2) are learning combination play against two opponents (O_1, O_2): A_1 pass the ball to A_2 through O_1 and O_2, and then A_2 shoot the ball (shown in Figure 2).
- There are two strategies (S_1, S_2).
  - S_1 : while A_1 pass through between O_1 and O_2 A_2 run forward to the point P,
  - S_2 : while A_1 pass beside O_2 A_2 stop waiting the ball.

In this situation, we make A_1 and A_2 learn the cooperative strategy for the opponents' various position pair. The kick power of A_1 and the dash power of A_2 are constant. Then, A_1 have to learn only the direction to kick, and A_2 have to learn only the number of times to dash.

3.2 Our Approach

This learning includes many issues that we mentioned in section 2.1. We approached for each problems as following.

- **Noisy environment**
  We use the neural network for the learning method, because it comparatively works well in a noisy environment.

- **Cooperation using the information from different standpoint**
  Neural network can cope with this problem, too. In this experiment, we tested whether it is possible or not.

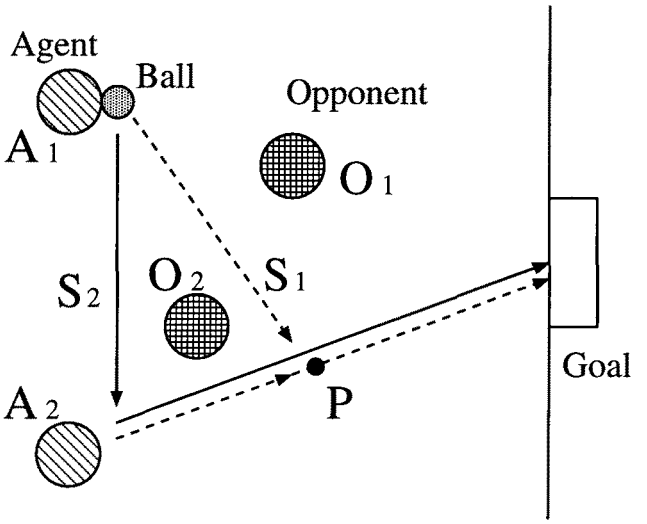

Fig. 2. situation used in this experiment

- **Limited field of vision**
  Now my agents use the last data for the lacking data. Machine Learning in such environment is one of the future works.

- **All agents have to select a same and correct strategy**
  In this experiment, agent try random action repeatedly. When the trial was successful, learn the pair of action and perceptional data using the back propagation.

The neural network for A_1 consists of 4 input-units, 50 first-hidden-units, 50 second-hidden-units and 20 output-units. And the neural network for A_2 consists of 4 input-units, 50 first-hidden-units, 50 second-hidden-units and 8 output-units. The inputs of these networks are relative direction and distance of O_1 and O_2. The number of output units means that, we separate the direction into 20 ways, and restrict the number of dash from 0 to 7.

One simulation is limited to 10 seconds. Then if the agents select different strategy, such as A_1 choose S_1 and A_2 choose S_2, they cannot shoot in time.

3.3 Results

In this experiment, we chose 5 square lattice points for possible position of each opponent. In this 625 situations, the agents learn about 100 random but success-

ful examples. This 100 examples include some examples in the same situations but ideal actions are different each other.

To learn these 100 examples completely, the neural network needed 10000 times of back propagation, for each of these 100 examples.

Table 1 shows the result of this experiment. Each column means the following.

- "goal": shoot action was success
- "near miss": pass seems succeeded but fail in shoot
- intercepted: intercepted by opponent
- failure: pass play failed at all

So, the sum of "goal" and "near miss" can be treated as the number of successful examples.

This result shows that the agents improved there success rate of pass play from 32% to 55%, with learning only 16% of noisy examples. Furthermore, A_1 learned that where is the best direction to kick the ball. Then A_2 could capture the ball at good place, then the number of the "goal" remarkably raised.

Therefore we can see that the agents could learn cooperative behaviors only with the perceptional information from their own stand point.

Table 1. Result of the experiment

	goal	near miss	intercepted	failure
before	34	165	98	328
	199 (32%)		426 (68%)	
after	126	215	88	196
	341 (55%)		284 (45%)	

4 Conclusion

The experiment shows the following things.

- Each agent can learn cooperative behaviors with the information from different stand points.
- After learning, agents can guess suitable answer for some situation which have not learned.
- Neural network is useful for the learning in such noisy environment.

Then, my agent can learn cooperative behaviors, if the function of the best direction and the special strategy are replaced with the neural network.

But many problems as followings are still remaining.

- In this experiment, the teammate does not move. If the agent learns for about a lot of situations of teammates, bigger neural network will be needed.

- The agent can not compensate for the lacking input. In this environment, the agent's field of vision is limited. So the lacking input will sometimes occur.
- This method takes a lot of time. Now it is far from adapting to cope with opponents in the game.

Therefore, main future works should be these three in this order.

- Experiment with changing the position of offensive agents. Then the agent can be used for soccer match.
- Deal with not only two against two but many kind of situation.
- Treat the situation, where sometimes lack the input data of neural network.

References

1. 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.
2. Itsuki Noda. "Soccer server: a simulator of robocup" In Proceedings of AI symposium '95, pages 29-34 Japanese Society for Artificial Intelligence December 1995.
3. Peter Stone and Manuela Veloso. "Multiagent systems: A survey from a machine learning perspective" IEEE Transactions on Knowledge and Data Engineering, June 1996.
4. Ming Tan. "Multi-Agent Reinforcement Learning: Independent Vs. Cooperative Agents" Proceedings of the Tenth International Conference on Machine Learning pages 330-337 June 1993.