Direct Reward and Indirect Reward in Multi-agent Reinforcement Learning

Masayuki Ohta

Cyber Assist Research Center
National Institution of Advanced Industrial Science and Technology
AIST Tokyo Waterfront 2-41-6 Aomi Koto-ku Tokyo 135-0064, Japan
ohta@carc.aist.go.jp

Abstract. When we apply reinforcement learning onto multi-agent environment, credit assignment problem will occur, because it is sometimes difficult to define which agents are the real contributors. If we praise all agents, when a group of cooperative agents get reward, some agents which did not contribute it will also reinforce their policies. On the other hand, if we praise obvious contributors only, indirect contribution will not be reinforced. For the first step to reduce this dilemma, we propose a classification of reward, and then investigate the feature of it. We treat a positioning task on SoccerServer for the experiments. The empirical results show that direct reward takes effect faster and helps obtaining individuality. On the contrary, indirect reward takes effect slower, but agents tend to form a group and obtain another effective positioning.

1 Introduction

When we apply reinforcement learning[5] onto multi-agent environment, problems peculiar to multi-agent environment arise. The problem that maximizing reward of individual agents does not guarantee maximizing the reward of the whole cooperative group of agents is one of them. We focus on this problem and are going to solve this by deciding whom to reward. Supposing the environment that many robots, which is unfamiliar to each other, are learning cooperative tasks without communication, we put the following constraints to the experiment environment. Supposing the environment that many robots meet the first time each other, we did our experiment under the constraints that agents learn simultaneously without communication in noisy environment. There are some researches which deal with similar environment. Sen et.al [8] showed that normal Q-learning[3] can acquire the optimal policy without sharing information by block pushing problem, and Arai[2] reported effectiveness of Profit-Sharing[4] in noisy multi-agent environment without communication, using pursuit Game. However, both of them are such kind of problem that maximizing reward of each agent lead to maximizing the reward as a whole automatically. In this paper we treat a positioning problem on simulated soccer, in which indirect helps are significant contribution as well as actions that achieves a goal directly.

G.A. Kaminka, P.U. Lima, and R. Rojas (Eds.): RoboCup 2002, LNAI 2752, pp. 359–366, 2003.

The outline of this paper is as following. In section 2, we introduce a problem of reinforcement learning on multi-agent environment, and define a classification of reward as this problem. In section 3, we show the details of our agents and the learning algorithm. In section 4, we show the result of experiments to investigate the feature of each kind of reward. And, in section 5, the related works are shown.

2 Problem of Multi-agent Reinforcement Learning

2.1 Reinforcement Learning in Multi-agent Environment

Reinforcement learning is a learning method that agents change their policies so as to maximize rewards given by the environment. We can apply this to a problem whose optimal action is not known, but we have to reward agents in proper condition. When a group of agents learns cooperative behaviors with reinforcement learning, the way to distribute the reward is one of the most important problems, which is called credit assignment problem. This is because indirect assists will be important factors as well as direct actions result in rewards. But it is sometimes difficult to define whose assists are worth rewarding, especially without sharing information, so a method that praise the whole agents is sometimes taken. Further, there is a following dilemma of whether to praise direct contributions or to praise all agents.

- If some agents which contributed directly only are praised, reinforced actions are relevant to the reward. But, all agents are going to achieve the goal only by itself, and after all, the group of agents will not maximize the sum of whole agents' reward. This is just like struggling for the reward with themselves.
- If all cooperative agents are praised when they achieved their goal, the combination of the agents' actions will be reinforced. But, because not all agents are relevant to the reward, some agents (such as agents just doing exploration) may reinforce bad policies.

It seems useful to use mixture of them in proper ratio, therefore, we have to investigate the feature of these reward beforehand.

2.2 Direct Reward and Indirect Reward

To distinguish between reward for direct contribution and reward for indirect contribution in multi-agent reinforcement learning, we call them "direct reward" and "indirect reward" respectively. The definitions are as followings.

- Direct Reward:
  The reward that was not provided if actions of other agents were not changed and only oneself selected other action.
- Indirect Reward:
  The reward that was provided even if actions of other agents were not changed and only oneself selected other action.

For example, in robotic soccer, direct reward is given to the player who succeeded in shooting a goal or simply kicked the ball, and indirect reward is given to the teammates who guarded opponents or defended their goal. Here, the point we should pay attention to is that the reward for the agent which passed the ball to the shooter is direct reward (even if it is called "assist"). This is because they could not shoot a goal without this action, obviously. The word of "direct reward" and "indirect reward" is used by Miyazaki[6] also. Their definitions are very similar to our definition, but different to some extent. In their definition, using an example of "Pursuit Game", direct reward is given to the agent which did the last action to catch the pray, and indirect reward is given to the other agents. But reward given to these surrounding agents is direct reward in my definition, because their last action is also necessary to catch the pray.

3 Details of Learning Agents

3.1 Learning Positioning on SoccerServer

As a testing-ground, we use SoccerServer[7] which is a simulator of robotic soccer. It realized abstracted real-world problem, and has been used for a lot of researchs. We investigated the feature of direct reward and indirect reward by experiment on learning effective positioning policy on SoccerServer. In soccer, positioning policy is closely related to the strength, and position for indirect contribution is very important as well as position for direct contribution. We consider an evolution of learning team against fixed positioning team, in which they completely have the same ability besides positioning. The effectiveness is measured by the score of the learning team. Both learning and fixed positioning agents are hand-coded, and behave as following.

- if (lost sight of the ball) look for the ball
- else if(ball is in kickable area) shoot or pass or clear (hand-coded)
- else if(nearest from the ball) chase the ball
- else go back to the base position and trace the ball

Each agent belongs to the learning team decides its base position with its own neural network. They divide the soccer field into a grid, then the neural network calculate the expected discounted reward in the case that their base position is at the center of the cell. The cell with the highest expected score in the neighborhood is selected as the next base position. The inputs of the neural network are "distance from the closest teammate", "distance from the closest opponent", "x-coordinate", and "y-coordinate" of the cell. Layout of players on the field is the state $s \in S$ of that time, and distance from teammates and opponents is the value that reflect the state. Also, the x-coordinate and y-coordinate can be treated as the next action $a \in A$ of the agent. Thus, the neural network is a function approximation of the action-value function $Q(s, a)$. The agents improve this function approximation in order to find an effective positioning.

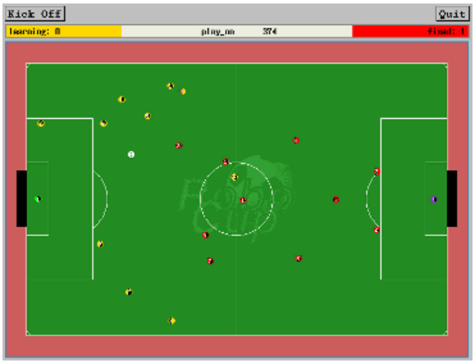

Fig. 1. A screen-shot: just after the learning experiment began. Learning team (left) against fixed positioning team (right)

3.2 Learning Algorithm

The purpose of the agent which belongs to the learning team is to find the optimal policy π, which is defined as the following.

$$Q^*(s,a) = \max_{\pi} Q^{\pi}(s,a) \qquad (\forall s \in S, \, \forall a \in A(s))$$

$$\begin{aligned} \text{where} \qquad & S = \text{All possible combination of all players' position} \\ & A = \langle (0,0),(0,1) \ldots (max\_x - 1, max\_y - 1) \rangle \end{aligned}$$

In this research, we adopt Monte Carlo method which is a kind of nonbootstrapping method to estimate $Q^{\pi}(s,a)$, because of its robustness against violations of the Markov property [10]. The expected return (because the memory is limited, we use the expected return in N steps) starting from s, taking the action a and thereafter following policy π is defined as

$$Q^{\pi}(s,a) \approx E_{\pi} \left\{ \sum_{k=0}^{N-1} \gamma^k r_{t+k+1} \,|\, s_t = s, a_t = a \right\}$$

where $0 \leq \gamma < 1$ is a discount-rate parameter, and r_x is the reward at time x.

Neural network of an agent, which approximate $Q^{\pi}(s,a)$, has random weight at first. Fig.1 shows a screen shot of the early stage of learning. In this figure, all players are putting position toward the edge of the field. Begin with this random policy, agents learn with algorithm shown in Fig.2. In the experiment using the SoccerServer, because it takes long time to change the base position, we execute 1. for each step, and from 2. to 5. for every 50 steps, indeed. And it is also an important point that, at the step 2. in the algorithm, the neural network is recalculated with the weights at that time before doing back propagation. Because of this, agents do not have to save the condition of the neural network in each step.

```
Initialize,
  initialize the neural network
  Histry ← empty list
  Reward_Sum ← empty list
  (Both size of History and size of Reward_Sum are $N$)

In each step,
  1. If (get reward $r_t$)
     From the bottom to the top $Reward\_Sum(n) \leftarrow Reward\_Sum(n) + \gamma^{n-1} \cdot r_t$
     where $Reward\_Sum(n)$ is the $n$th element of the Reward_Sum from the bottom, and $0 \leq \gamma < 1$ is a discount-rate parameter.

  2. Take out the top element of History($N$) and Reward_Sum($N$), then reinforce as the following equation using back propagation.
       $Q(History(N)) \leftarrow (1 - \alpha)Q(History(N)) + \alpha \cdot Reward\_Sum(N)$
     where $0 \leq \alpha < 1$ is a learning rate factor.

  3. Shift all elements of the History and Reward_Sum to the top direction.

  4. Recalculate the best action.
     $\begin{cases} probability\ 1-\epsilon & a_t \leftarrow \max_a Q^{\pi_t}(s,a) \\ otherwise & a_t \leftarrow random\ a \in A \end{cases}$
     where $0 < \epsilon \leq 1$ is probability of exploration

  5. Add the current state-action pair to the bottom of the History.
     History(1) ← $(s_t, a_t)$
     Reward_Sum(1) ← 0
```

Fig. 2. Learning algorism in this experiment

4 Experiments

In order to examine a difference of the feature between direct reward and indirect reward, we did learning test with each reward separately. In each experiment, we continue executing the soccer simulation of learning team and fixed position team, and the effectiveness is measured by the score of the learning team in a unit time. The difference between the learning team by direct reward and the learning team by indirect reward is only the condition of reward. In both case, we use discount-rate $\gamma = 0.9$, learning rate $\alpha = 0.5$, and probability of exploration $\epsilon = 0.1$.

4.1 Learning with Direct Reward

In the first experiment, agents learn a policy which is effective against a fixed position team, with direct reward only. We gave agents the direct reward when

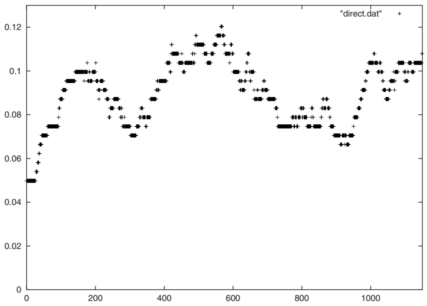

Fig. 3. Score of the learning team reinforced only by direct reward against a team with fixed policy

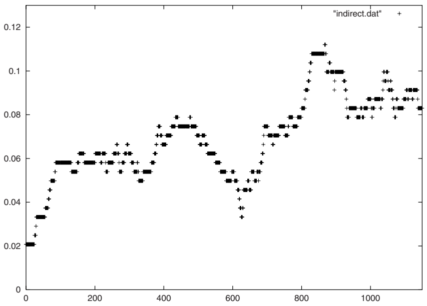

Fig. 4. Score of the learning team reinforced only by indirect reward against a team with fixed policy

they kick the ball by themselves. Fig.3 shows the learning curve in which x-axis indicates the time scale by a minute (= 600 step), and y-axis indicates the score of learning team in a minute. The score is the average value in one hour. Because of the randomness produced by the simulator, the learning curve is not stabilized, but we can see the agent could learn effective positioning. In this experiment, all agents expected to take close position each other where they can kick the ball frequently, but the experiment shows that agents took relatively distributed positioning. This is because only one agent can kick the ball in the same time and therefore learned having individuality. A typical pattern which is acquired by the learning team is like Fig.5.

4.2 Learning with Indirect Reward

Second, we did just the same experiment as the first one, except for the reward. In this experiment, agents get reward only when they see a teammate kicking the ball. The result of this experiment is shown in Fig.4, and a typical positioning pattern is like Fig.6. The learning went much slower than the experiment with the direct reward, but at last, agents shows almost the same efficiency. In this case, agents acquired a positioning policy that all agents tend to form a group, and moves together. Even if this indirect reward teaches that "one player is enough to chase the ball", the result shows the opposite outcome. This is only because of the feature of the SoccerServer on which agents can not discriminate teammates if they are away from each other. With this feature, because some agents chase the ball and others follow them, this learning team changes its position by the position of the ball.

4.3 Discussion

In the above two experiments, we can see the following results.

- Direct reward affects learning very quickly, and indirect reward affects learning relatively slow.

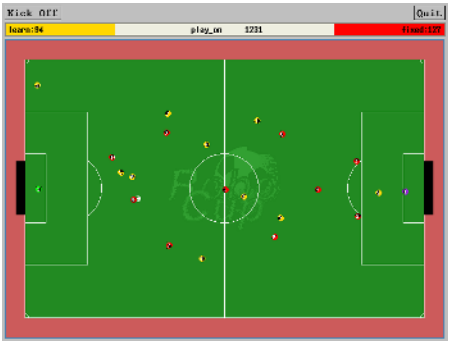

Fig. 5. Typical positioning acquired by direct reward

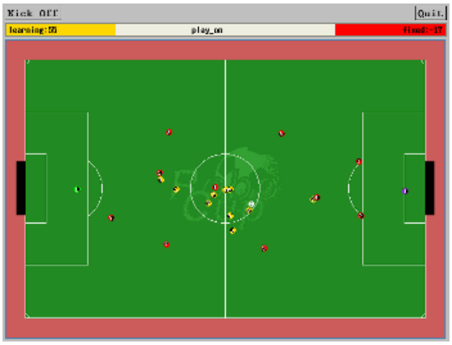

Fig. 6. Typical positioning acquired by indirect reward

- Both agents showed the same effectivity finally, but they acquired totally different policy.
- Agents learned with direct reward took distributed formation, and the agents learned with indirect reward took crowd formation.

In multi-agent environment individuality is very important factor, because that is related to distribute the role. But, in this experiment, the other acquired policy also had the similar effectivity, even if they are totally different. Therefore, mixture of these two kind of rewards is worth testing. Especially, if ”direct reward” is used in a big ratio at first, and if it is reduced along the time we can expect that the score during the learning can be improved.

5 Related Works

Observational Reinforcement Learning[1] solved the problem that exploration causes bad effect on the other agents' learning in multi-agent environment. They also treat a positioning task on SoccerServer. Their solution is to use a reward which reinforce the position where the agents think it is good, by their perception information. Using this rewarding agents can have almost the same effect as the exploration, even if it reduce their exploration rate. This method is very useful, but it is a kind of supervised learning, and agents need to know some candidate answers beforehand. It is also interesting that, the agents reinforced by observational reward have similar feature to the agents reinforced by indirect reward.

Miyazaki[6] also classify the reward for reinforcement learning into direct reward and indirect reward. But as we already pointed out, the definition is different from ours. They showed necessary and sufficient condition of direct and indirect reward ratio to preserve the rationality. In their research, they are interested in the fixed ratio, but in this paper we focused on the change along the time.

6 Conclusion

We proposed a classification of reward, direct reward and indirect reward, in reinforcement learning. The direct reward affects quickly, and indirect reward affects slowly. Individuality is acquired using direct reward, but crowd formation acquired with indirect reward also showed the similar effectivity. We could see the trade-off between these reward, therefore, changing the mixture ratio of direct reward and indirect reward during the learning can be a future work.

References

1. T. Andou. Andhill-98: A robocup team which reinforces positioning with observation. In M. Asada and H. Kitano, editors, *RoboCup-98: Robot Soccer World Cup II*, pages 338–345. Springer Verlag, 1998.
2. S. Arai and K. Sycara. Effective learning approach for planning and scheduling in multi-agent domain. In *Proceedings of the 6th International Conference on Simulation of Adaptive Behavior*, pages 507–516, 2000.
3. C.J.C.H.Watkins. *Learning from Delayed Rewards.* PhD thesis, Cambridge University, Cambridge, England, 1989.
4. J. J. Grefenstette. Credit assignment in rule discovery systems based on genetic algorithms. In *Machine Learning*, volume 3, pages 225–245, 1988.
5. L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.
6. K. Miyazaki and S. Kobayashi. Rationality of reward sharing in multi-agent reinforcement learning. In *Second Pacific Rim International Workshop on Multi-Agents*, pages 111–125, 1999.
7. I. Noda, H. Matsubara, K. Hiraki, and I. Frank. Soccer server: A tool for research on multiagent systems. In *Applied Artificial Intelligence*, volume 12, pages 233–250, 1998.
8. S. Sen, M. Sekaran, and J. Hale. Proceedings of the 12th national conference on artificial intelligence. In *Learning to Coordinate without Sharing Information*, pages 426–431, 1994.
9. P. Stone and D. McAllester. An Architecture for Action Selection in Robotic Soccer. In *Proceedings of the Fifth International Conference on Autonomous Agents*, 2001.
10. R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, 1998.