Reinforcement Learning for Soccer Multi-agents System

Conference Paper · January 2009

DOI: 10.1109/CIS.2009.275 · Source: DBLP

2 authors:

Fahimeh Farahnakian
University of Turku

Nasser Mozayani
Iran University of Science and Technology

Reinforcement Learning for Soccer Multi-agents System

Fahimeh Farahnakian
School of Computer Engineering
Iran University of Science &Technology
Tehran, Iran
Farahnakian@comp.iust.ac.ir

Nasser Mozayani
School of Computer Engineering
Iran University of Science &Technology
Tehran, Iran
mozayani@iust.ac.ir

***Abstract*— Recently the reinforcement learning method is actively used in multi-agent systems. Because of this method played a significant role by handling the inherent complexity of such systems.**

Robotic soccer is a multi-agent system in which agents play in real-time, dynamic, complex and unknown environment. Since the main purpose of a soccer game is to score goals, it is important for a robotic soccer agent to have a clear policy about whether it should attempt to score in a given situation.

Therefore we use reinforcement learning for optimizing policy. In the proposed method, the state spaces include two important parameters for shooting toward the goal; the distance between the ball and the goalkeeper and the probability which is obtained from the research of the UvA team. Of course, we select these parameters for effective features of scoring. Because they are more effective learning algorithm in real-time simulated soccer agent [1].

Experimental results have shown that policy achieved from reinforcement learning lead to more effective shoots toward the goal in simulated soccer agent.

Keywords-Reinforcement learning; Multi-agent system; Q-learning; Robocup

I. INTRODUCTION

Reinforcement learning [2] is known as a good machine learning method to construct optimal policies in unknown environments. Also, RoboCup is a standard problem so that various theories, algorithms and architectures can be evaluated [3]. Behavior learning for complex tasks is also an important research area in RoboCup.

In our previous describes the use of decision tree to kick and catch the ball for two simulated soccer agents [4].

In this paper, we propose a reinforcement learning method called Q-learning [5]. We apply the proposed method for learning the kicking skill of shooter player acting in the RoboCup 2D simulator. Many parameters affect the result of shooting toward the goal. Simulation teams such as UvA Trilearn [6], TsinghuAeolus [7], etc. have good techniques for scoring behavior.

In the proposed method, we focus two effective parameters:

-The distance between the ball and the goalkeeper

- The best point of the goal and the probability of scoring at this point are calculated by UvA Trilearn simulation team [8].

In section 2, we introduce the used reinforcement learning algorithm. Section3 presents Implementation and practical results and finally, section 4 is the conclusion.

II. Q-LEARNING

The standard Q-learning algorithm has several stages [9]:

1. for each s and a, initialize the table entry Q(s, a) to zero

2. Observe the current state (s)

3. Do forever:

- Select an action (a) through one of these methods and execute it:

- *Exploration or random*
- *Exploitation or based on Q-table*
- *Receive reward (r)*
- *Observe the new state (s')*
- *Update the table entry for Q(s, a) as follow:*

$$Q(s,a) = r + \gamma \max_{a'} Q(s',a'))$$

- $s \leftarrow s'$

There are two methods for selecting one action from the possible actions in every state [7 and 2]:

Exploration: or random action selection. So, optimal actions, which are not chosen yet, are added to the table.

Exploitation: action selection is according to the learned Q-table.

It is clear that action selection is more exploration at the beginning of learning, and is more exploitation towards the end of learning.

III. SIMULATION RESULT

In order to apply the Q-learning for scoring problem, we consider two steps. One step is train step that shooter player interactions with the world and Q-table is gathered. The other step is test step, shooter player that is using the

978-0-7695-3931-7/09 $26.00 © 2009 IEEE
DOI 10.1109/CIS.2009.275

"knowledge" of the trained policy to determine the best action. Figure 1 shows how the different parts interact together.

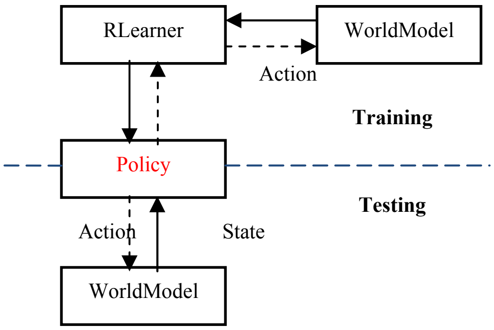

Figure 1. Q-learning implementation

The RLearner include Q-learning, the output of this algorithm is a Q-table, which includes learned Q-Values when an agent executes action a in state s.

The WorldModel holds the current state and provides feedback about the next state, the validity of a certain action and the reward for a certain action. The policy links it all together. It is updated by the RLearner based on the experience it is having with the world and later on consulted by the test part for choosing an optimal action.

The action spaces contained “to shoot toward the goal” or “not to shoot toward the goal”.

Since parameters in state space affect the result of a shooting toward the goal. Thus, the state space is represented by internal information the learning agent maintains such as the distance between the ball and the goalkeeper and the probability which is obtained from the research of the UvA team. A reward is given to the shooter player is +1 if the result shoot is goal.

A program for gathering training data and the Q-learning algorithm program were written in C++ language. In our simulations, one trial ends when the shooter player can successfully shoots to goal or the maximum time step is reached. We specified the maximum time step as 200 simulator cycles. We performed the simulation for 1000 trials. Every 200 trials we examined the performance of the Q-learning by calculate number of goals.

IV. CONCLUSION

We show simulation results of the Q-learning in figure 2, figure 3and figure 4. In figure 2, the horizontal axis displays percentage of the number of goals divided by the number of times that shooter kick to the goal. We can see that the number of goals increases as the number of trials increases. Also, the numbers of goals in the train step are more than in the test step. In addition, figure 3 indicate the average of Q-values on 1000 trails. The Q-learning curve in figure 3 did not monotonically increase to the number of trials. This is because the RoboCup server employs some degree of noise in the movement of objects such as the ball and agents and in the perception such as the information on the location and velocity of the objects. Also we observed the effect of the result of Q-learning at the previous trials on the performance of succeeding learning.

Also, figure 4 shows a comparison between UvA shooter and the shooter player trained with Q-learning in this paper.

In general in this research we applied the Q-learning for scoring problem. Shooter player select two action “to shoot toward the goal” or “not to shoot toward the goal” when the ball is kickable margin.

Simulation results showed that the learning agent can gradually learn to successfully kick ball toward the goal .In comparison between UvA shooter and the learning agent indicated the number of goals increase. Also, that these parameters affect the result of a shooting toward the goal. Thus we can have better scoring behavior by considering them.

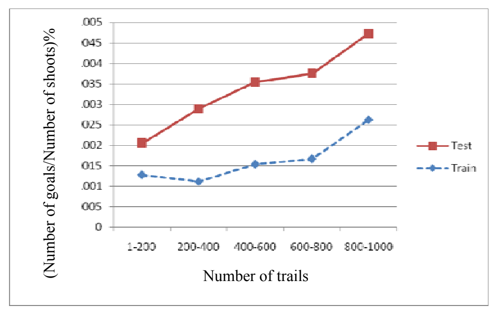

Figure 2: Number of goals in Train step and Test step

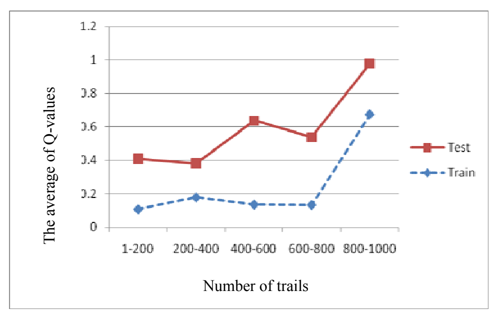

Figure 3: The average of Q-values in Train step and Test step

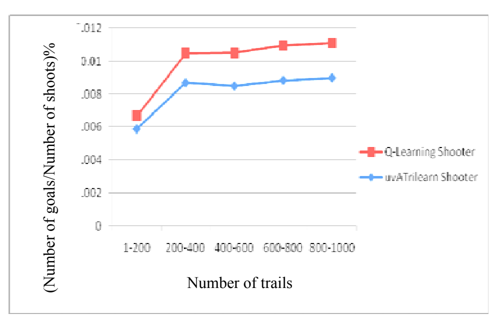

Figure 4: Number of goals in UvA shooter and Learning shooter

REFERENCES

[1] F.Farahnakian and N.mozayani, "Evaluating Feature Selection Techniques in Simulated Soccer Multi gents System" , International Conference on Advanced Computer Control,2009,IEEE,Singapore.

[2] R. S. Sutton and A. G. Barto, "Reinforcement Learning: An Introduction", MIT Press, 1998.

[3] Coradeschi S., Noda I., Stone P., Osawa E. &Asada M., "RoboCup Synthetic Agent Challenge", 1997.

[4] F.Farahnakian and N.mozayani,"Learning through Decision Tree in Simulated Soccer Environment", Computational Intelligence and Security, 2008. , IEEE, Volume: 2, On page(s): 68-70, China.

[5] C. J. C. H. Watkins and P. Dayan, "Q-Learning", Machine Learning, Vol. 8, pp. 279–292, 1992.

[6] Kok J., Vlassis N. & Groen F., "UvA Trilearn 2003 Team Description", Faculty of Science,University of Amsterdam, 2003.

[7] Jiang C. & Jinyi Y., "Architecture of TsinghuAeolus", TsinghuAeulos 2001 Team Description Paper. In Proceedings of RoboCup-2001: Robot Soccer World Cup V, 2001.

[8] Kok J., Boer R., Vlassis N. & Groen F., "Towards an Optimal Scoring Policy for Simulated Soccer Agents", Faculty of Science, University of Amsterdam, (2002).

[9] Mitchell T.M., Machine Learning, McGraw-Hill Press, International Edition, (1997).