Team Description Paper GPR-2D 2012

José Rodrigo F. Neri[1], Maicon R. Zatelli[1], Carlos H. F. Santos[2], João A. Fabro[3]

[1] Department of Automation and Systems (DAS), Federal University of Santa Catarina (UFSC), Florianópolis, SC, Brazil
{jrneri,,maicon}@das.ufsc.br
[2] Robotic Research Group (GPR), Center of Engineering and Exact Sciences (CECE), State University of West of Paraná (UNIOESTE), Foz do Iguaçu, PR, Brazil
chf.santos@uol.com.br
[3] Computational Intelligence Research Group, Informatics Department (DAINF), Federal University of Technology of Paraná (UTFPR), Curitiba, PR, Brazil
fabro@utfpr.edu.br

Abstract. This document describes a simulated robotic soccer team (GPR-2D). The team is implemented using an Artificial Intelligence (AI) strategy to modify the behavior of players of one team of soccer robots of the RoboCup 2D simulation category. This strategy used is a Reinforcement Learning (RL) algorithm, Q-learning. This algorithm define how each agent should act according to the characteristics of the opponent players, selecting the best action to take in a given state of the environment.

Keywords: Robot Soccer, Reinforcement Learning, Q-learning, Robocup.

1 Introduction

The robot soccer was a problem proposed by an international group of researchers in artificial intelligence (AI) and intelligent robotics in 1996 [2].

The main features in a single football team of robots are: each player must identify objects in the environment as the ball, and the robots that make up your team and the opposing team. Each player should be able to represent the environment, select goals, plan and implement actions to achieve the goals of the team [3]. As these goals depend upon individual decisions taken by each player, the paper proposes the use of a Reinforcement Learning algorithm (Q-Learning) to modulate the decision process of each simulated player, based on a training section that evaluates the decisions that best suited the players in previous matches. This paper thus describes the training of the GPR-2D team, of the RoboCup 2D simulation category.

This paper has six sections. The first section present the problem and the motivation for implementing the proposal. Section 2 presents the GPR-2D team that is based on the Helios team from Japan. The third section describes the reinforcement learning method implemented (Q-Learning). Section 4 presents the modeling of the problem.

In section 5 we show the implementation of the algorithm. In the final section the results are discussed, as well as the conclusions and some ideas that can serve as directions for future research are presented.

2 The GPR-2D Team

The GPR-2D team is the current champion of the Robocup Open Brazil 2011[6] and currently developed by GPR (Grupo de Pesquisas em Robótica - Robotics Research Group) from the State University of West-Paraná(UNIOESTE), in collaboration with the Federal University of Technology of Paraná (UTFPR) and the Federal University of Santa Catarina (UFSC) (all from Brazil). The team is based on the source code of the Helios team (Japan) that participated at the Robocup 2011. The main advance proposed is the modification of the behavior of each player using the Q-learning [5], a Reinforcement Learning algorithm. This strategy allows the agents to act appropriately when they are in possession of the ball, picking the best action to take in a given state of the environment.

2.1 The Helios 2011 Base Team

Helios is one team from Japan that released the source code of its 2D simulation team (agent2D). Since its appearance, new tactical skills and optimizations have been made to this team, and thus it has become base for many teams, including ours.

3 Reinforcement Learning

According to [5], Reinforcement Learning (RL) is a formalism that allows the AI agent to learn from its interaction with the environment in which it is inserted. The main objective of the algorithm is to find the best rewarding action among all possible actions in any situation. Learning takes place through selection of a certain action when in a certain situation, and evaluating the outcome. The outcome is then used to reinforce the probability of taking certain actions, or decrease the probability of taking other actions. There are several techniques of RL, one of the most used is the Q-learning algorithm, it is also used in related works on simulated robots soccer on Robocup [7-9].

3.1 Q-learning

The Q-learning method can be applied when an agent is embedded in a finite and discrete world. At each iteration, the agent observes the current state of the environment, choose between a finite number of choices, and receives an immediate reward for each of its actions [1].

In its simplest form, Q-learning with one step can be defined by the following expression:

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + a[rt+1+\gamma \max_a Q(s_t+1,a) - Q(s_t,a_t)] \quad (1)$$

- where s_t corresponds to the current state;
- a_t is the action taken at state s_t;
- r_t is the reward received by taken the action a_t at the state s_t;
- s_t+1 is the next state;
- γ is the discount factor *(0<= γ <=1);*
- α is the learning rate *(0 <= α <1)*.

The function $Q(s_t,a_t)$ is the value associated with the state-action pair (s_t,a_t) and represents how good is the choice of this action in maximizing the cumulative return function. The action-value function $Q(s_t,a_t)$, that store the reinforcements received, is updated from its current value for each state-action pair. Thus, a part of the reinforcement received in a state will be transferred to the state prior to this.

In Q-learning algorithm, the choice of action to be performed in a given state of the environment can be made with any criteria of exploration / exploitation, including randomly. A policy that is a widely used is called ϵ-greedy, where the agent can choose a random action, or action that has the largest value increase in $Q(s_t,a_t)$ [4]. The choice with the ϵ-greedy policy occurs as follows:

$$\begin{cases} a_{random} & if \;\; q \le e \\ \max\limits_{a_t} Q(s_t,a_t) & otherwise \end{cases} \quad (2)$$

If the value of Q chosen at random is less than the value set, a random action is selected, otherwise the action in table $Q(s_t,a_t)$ with the largest reinforcement value assigned is selected.

4 Problem Modeling

To apply the Q-Learning algorithm, one of the first steps is the discretization of the states that each simulated player can achieve. The proposed states for this problem were:

1. **Lead_pass:** When in this state, it is possible to launch the ball to a fellow player that is better positioned;
2. **AdversaryBehind:** The closest adversary player is behind the player that is in possession of the ball;
3. **AdversaryFarAway:** The distance to the closest adversary is bigger than 7 meters;
4. **AdversaryNotSoClose:** The distance to the closest adversary is bigger than 6 meters;

5. **AdversaryIsClose:** The distance to the closest adversary is bigger than 5 meters;
6. **AdversaryVeryClose:** The distance to the closest adversary is bigger than 4 meters;
7. **KickOpportunity:** There is a direct line from the player that possesses the ball and the adversary goal.

The following are the possible actions that can be taken by each agent:

1. **Launch:** Launch the ball towards a fellow player better positioned, i.e., that is much closer to the adversary goal;
2. **SlowForward:** To slowly advance with the ball;
3. **FastForward:** To quickly advance with the ball;
4. **Pass:** To pass the ball to a fellow player that is close, but not necessarily best positioned;
5. **Dribble:** Try to dribble the closest adversary player;
6. **Hold_the_ball:** Just hold the ball;
7. **Kick:** To kick the ball in the direction of the adversary goal.

Seven states were defined for the environment, and seven actions for the agents.

After defining the states of the environment and actions of the players, it is necessary to define the matrices R (Reinforcement Matrix) and Q. The matrix R is where the reinforcements will be assigned, which will define the actions of the agent.

In the definition of the matrix R (Table 1), the lines are the states of the environment and the columns are the players' actions. Each state of the environment can have one or more actions. A value of 0 (zero) means that this action has no reinforcement, -1 (minus one) means that the action is not performed, and values greater than zero means reinforcement for that action. It is observed that for state 7 (KickOpportunity) the agent can choose the action 7 (Kick) in which a reinforcement is assigned 100.

Table 1. Matrix R.

State\ Action	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	0	-1	0	0	-1	-1	-1
2	-1	0	0	0	-1	-1	-1
3	-1	0	0	-1	-1	-1	-1
4	-1	0	0	0	-1	-1	-1
5	-1	-1	-1	0	0	0	-1
6	-1	-1	-1	0	0	0	-1
7	-1	-1	-1	-1	-1	-1	100

In the Q matrix the values of $Q(s_t, a_t)$ assigned by Q-learning algorithm are stored. At first, the matrix has only zeroes, but at each new game, the values obtained at the end of the previous game are maintained.

5 The Q-Learning Algorithm

The implemented Q-learning algorithm is as follows:

```
Do
  Observe actual state s_t
  Choose an action that will be executed according to the
  e-greedy policy
  Execute the selected action
  If there is change of state in relation to previous cy-
  cle, then:
  Determine the reinforcement to be assigned to previous
  state
  Update Q(s_t, a_t) with the reinforcement value using
  Q(s_t, a_t) ← Q(s_t, a_t) + a[rt + 1 + γ max_a Q(s_t + 1, a) − Q(s_t, a_t)]
  Save the actual state to verify later
While the game doesn't end
```

Three interest areas were defined on the field, as can be seen on fig. 1.

Fig. 1. Shows the three different interest areas (each one has different Q-Learning Matrices).

For each of these areas were defined a set of Q and R matrices. If the agent is inside the area labeled as 2, for example, it uses the values of R2 and Q2 matrices. If the agent moves to the area labeled 1, it uses the values of matrices R1 and Q1. Depending on the position of the agent in the field, it will behave differently, even if it is in the same state of the environment.

6 Conclusions

This paper has presented a proposal to use the Q-learning algorithm to provide a learning mechanism for simulated robotic agents of the RoboCup 2D simulation category. By this learning mechanism, these agents have shown improved behavior, adapting themselves individually to the different situations presented to them, in order to maximize the global performance of the team in scoring goals.

For future work in the area, we suggest the use of Q-learning method also for players without ball possession, making these agents may have a better positioning in the field, both in the center of the field or in defense positions, preventing the opposing team from scoring goals, even send or receive a pass. It is also suggested a study of further refinement of the states of the environment and the actions that can be taken by the simulated robot soccer players. The authors plan to apply this approach to other Robocup categories, such as 3D Simulation and Small Size.

References

1. Dayan, P. Technical Note Q-learning, Centre for Cognitive Science, University of Edinburgh, Scotland: University of Edinburgh, (1992).
2. Kitano H., Asada M., Kuniyoshi Y., Noda I.,. “The robocup synthetic agent challenge, 97”. International Joint Conference on Artificial Intelligence (IJCAI97), Nagoya, Japan (1997).
3. Neri, J. R. F; Santos C. H. F.; Fabro, J. A. Sistema Especialista Fuzzy para posicionamento dos jogadores aplicado ao Futebol de Robôs. IX Simpósio Brasileiro de Automação Inteligente (SBAI). Brasília - DF, (2009) (in portuguese).
4. Rosa, Allan P. Aplicações Especiais para Futebol de Robôs 2D Simulado. Itajubá - MG: UNIFEI, (2008).
5. Sutton, R. S.; Barto, A. G. Reinforcement Learning: An Introduction. Massachusetts: MIT Press, Cambridge, (1998).
6. Brazilian Robotics Competition Results Page, http://www.cbr2011.org/Sim2D.htm, Accessed: 01/03/2011.
7. Farahnakian, F.; Mozayani, N.;,"Reinforcement Learning for Soccer Multi-agents Sytem," Computational Intelligence and Security, 2009. CIS '09. International Conference on, vol.2, no., pp.50-52, 11-14 Dec. (2009).
8. Li Xiong, Chen Wei, Guo Jing, et al.A new passing strategy based on Q-learning algorithm in RoboCup, Computer Science and Software Engineering, Wuhan, Hubei, 12-14 Dec (2008), 1 :524-527.
9. Azam Rabiee and Nasser Ghasem-Aghaee, A Scoring Policy for Simulated Soccer Agents using Reinforcement Learning, 2nd International Conference on Autonomous Robots and Agents, New Zealand, (2004).