2nd International Conference on Autonomous Robots and Agents
December 13-15, 2004 Palmerston North, New Zealand

A Scoring Policy for Simulated Soccer Agents Using Reinforcement Learning

Azam Rabiee
Computer Science and Engineering
Isfahan University,
Isfahan, Iran
azamrabiei@yahoo.com

Nasser Ghasem-Aghaee
Computer Science and Engineering
Isfahan University,
Isfahan, Iran
aghaee@eng.ui.ac.ir

Abstract

The robotic soccer is one of the complex multi-agent systems in which agents play the role of soccer players. The characteristics of such systems are: real-time, noisy, collaborative and adversarial. Because of the inherent complexity of this type of systems, machine learning is used for training agents. 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 s/he should attempt to score in a given situation. Many parameters affect the result of shooting toward the goal. UvA Trilearn simulation team considers two more important parameters for this behavior. This paper describes the optimizing policy which is used in the UvA team, by choosing two additional important parameters as well as using reinforcement learning method.

Keywords: multi-agent system, RoboCup soccer simulation league, simulated soccer agents, scoring policy, reinforcement learning, Q-learning

1 Introduction

Multi-agent system is a complex system involving multiple agents in which each agent has independent and different agent's behavior. When a group of agents in a multi-agent system share a common long-term goal, they can be said to form a *team*. The agents in the environment that have goals opposed to the team's long-term goal are the team's *adversaries*.

The robotic soccer is one of the complex multi-agent systems in which agents play the role of soccer players. The main goal of robotic soccer is to have a perfect domain for researchers and a standard problem for investigating and examining new artificial intelligence as well as multi-agent approaches and techniques [1]. This scientific game provides a domain to examine artificial intelligence ideas, versus other ideas.

Features of the robotic soccer domain are: real-time, noisy, collaborative and adversarial [2]. Firstly, real-time domains are those in which agents should response quickly to a dynamically changing environment. Second, noisy domains are those in which agents cannot accurately perceive the world, nor can they accurately affect it. Third, collaborative domains are those in which a group of agents share a common goal and finally, adversarial ones are those in which there are agents with competing goals.

RoboCup[1], competition of soccer teams has several leagues: simulation league [3], E-level[2] robot, small size robot, middle size robot, 4-legged robots and humanoid league. Simulation league involves three classes: 2D, 3D and coach.

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 he should attempt to score in a given situation. In this paper, we focus on the scoring behavior of 2D simulated soccer agents.

The mass of parameters, noise, uncertainty or existence of inaccurate information complicate the problem further. Because of the inherent complexity of the agent's behaviors in this type of systems, machine learning is used for training more than other similar techniques. Many parameters affect the result of shooting toward the goal. Simulation teams such as UvA Trilearn [4], TsinghuAeolus [5], etc. have good techniques for scoring behavior.

UvA Trilearn team, which has been one of the best simulation teams for several years, was the champion of the world in RoboCup 2003 with the highest number of goals. This team considers two more important parameters for scoring behavior. In this paper, we add two other effective parameters and train it by what is called reinforcement learning method.

In section 2, we introduce the used reinforcement learning algorithm. In section 3, UvA Trilearn scoring policy is reviewed, then two other parameters and used training data is described. Section 4 presents Implementation and practical results and finally, section 5 is the conclusion.

[1] Robot World Cup Initiative
[2] Entry-level

2 Reinforcement Learning

The target of reinforcement learning algorithms is to find the best and the most optimal action in every state of environment. In these algorithms, agents perceive the state of the environment, select one of the possible actions and execute it; at the end of which they receive immediate rewards from the environment (figure 1). The main goal is, thus, to find the best sequence of actions which yield the maximum summation of rewards [6][7].

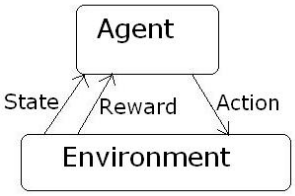


Figure 1: The interaction of agent and environment

Reinforcement learning algorithms are various because of [7]:

- Online or offline learning
- Deterministic or non-deterministic reward functions
- Existence or absence of indirect and delayed rewards
- Deterministic or non-deterministic action effects

Standard reinforcement learning is an online learning algorithm which is capable of learning control policy for MDP[1], based on long-term delayed rewards and deterministic reward functions and action effects.

The output of this algorithm is a Q-table, as shown in figure 2, which includes learned Q-Values when an agent executes action *a* in state *s*.

State	Action	Q(s, a)

Figure 2: Q-Table

The standard reinforcement learning algorithm has several stages [6]:

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:*
     $$Q(s,a) \leftarrow r + \gamma \max_{a'} Q(s',a')$$
   - $s \leftarrow s'$

Each time, this algorithm starts with an initial state and reaches a goal state by executing a sequence of actions and receiving rewards. Each sequence, which starts from an initial state and ends with a goal state, is named an *episode*.

In every goal state, by executing any action the agent comes back to the same state and receives no reward. These states are named *absorbing* states. The iteration loop is repeated by the number of gathered training data. There are two methods for selecting one action from the possible actions in every state [6][7]:

- *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.

3 Scoring Behavior

Scoring behavior is the most effective one in the result of a game; thus, it is important to have a clear policy for scoring goal. UvA Trilearn simulation team has one of the best scoring techniques [8]. In this technique, the best point of the goal and the probability of scoring at this point are calculated. If the probability of goal is greater than a threshold, agent shoots toward the goal point; otherwise, the agent executes another action.

The scoring event can be decomposed into two independent events:

- The ball not passing the goalposts
- The ball passing the goalkeeper

Thus, the probability of scoring is equal to multiplication of the probability of these two independent events.

If the player shoots straight towards the goal (figure 3) and by considering the *Central Limit Theorem,* the probability of the first event has a Gaussian distribution. This theorem states that under certain conditions the distribution of the sum of N random variables will be Gaussian as N goes to infinity.

This Gaussian distribution has a zero mean and a standard deviation $\sigma(d)$. This deviation, based on the distance to goal, is calculated by gathering 1000 training data. It can be computed as equation (1).

[1] Marcov Decision Process

$$\sigma(d) = -1.88\ln(1 - d/45) \qquad (1)$$

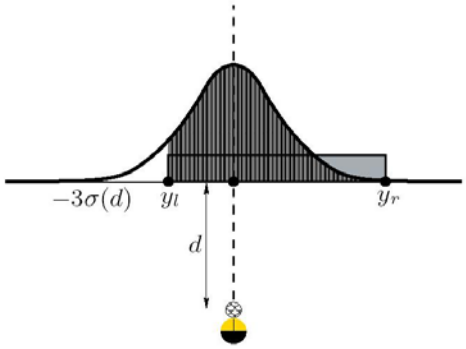


Figure 3: Shooting straight to the goal

When a player shoots at an angle toward the goal line (figure 4), the probability that the ball enters the goal is equal to one minus the probability that the ball goes out from the left goalpost minus the probability that the ball goes out from the right goalpost; in other words, it is equal to one minus the sum of the shading area of figure 4.

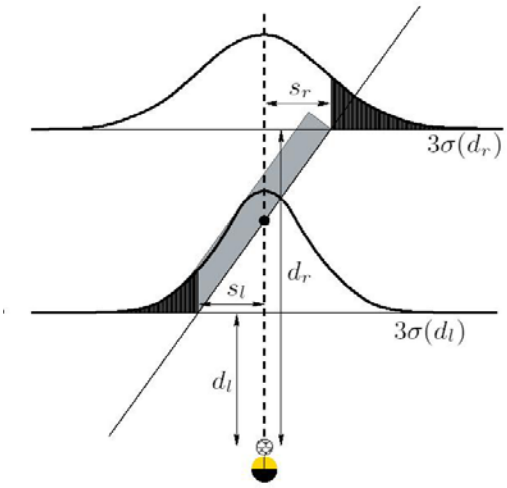


Figure 4: Shooting at an angle to the goal

To calculate the probability of the second event, passing the ball from the goalkeeper, the team considered two relevant parameters (figure 5):

(1) The absolute angle a between the goalkeeper and the shooting point,

(2) The distance d between the ball and the goalkeeper.

This sub-problem can be defined as a classification problem. These two values form a two-dimensional feature vector on which the classification has been based.

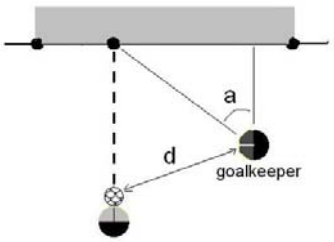


Figure 5: The two parameters, *a* and *d*

For computing the *posterior* probability associated with the prediction of each class, a data set was formed by gathering 10,000 training data. This data set shows that there is an almost linear discriminant function between the two classes which are passing the goalkeeper or not passing.

According to this data set, the almost linear discriminant function is defined in equation (2) and the probability of the second event can be computed as equation (3).

$$u = (a - 26.1) * 0.043 + (d - 9.0) * 0.09 - 0.2 \qquad (2)$$

$$P(pass\_goalkeeper \mid u) = \frac{1}{1 + \exp(-9.5u)} \qquad (3)$$

Finally, the probability of scoring goal is computed by multiplication of the probability of "the ball not passing the goalposts″ and the probability of "the ball passing the goalkeeper″. As we mentioned above, if the probability of goal is greater than a probability threshold, the agent shoots toward the goal; otherwise, there is little chance of a goal being scored, thus, he does another action.

In this research, the UvA scoring method is implemented and a set of 40,000 sample data contains the scoring probability and the result of shoot is gathered versus the goalkeeper of the UvA 2003 team.

The probability threshold considered to be 0.9. Figure 6 shows the performance of the UvA scoring method versus different probability threshold values from 0.9 to 1. Here the performance means the number of goals divided by the total number of shoots towards the goal.

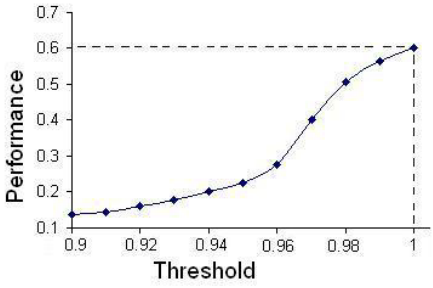


Figure 6: UvA scoring results

The maximum performance is around 60% in threshold 1. Thus, experiments show that the result of

the UvA scoring is not enough and could be optimized.

In addition to the two proposed parameters, angle and distance, there are other features which also affect the result of a shooting toward the goal, like body and neck angles of the goalkeeper and the number of teammates and adversary players around the goal and etc.

4 Implementation and Practical Results

In this paper, we consider two parameters, the body and the neck angle of the goalkeeper, beside the probability which is obtained from the research of the UvA team. A set of 40,000 training data is gathered versus the goalkeeper of the UvA 2003 team for training the scoring behavior via Q-learning.

Each of these examples includes three features:

- Probability
- The body angle of the goalkeeper
- The neck angle of the goalkeeper related to his body angle

These features indicate *initial* states. The results of shooting, which are one of the *goal*, *save* and *out*, are *target* states.

Again the probability threshold was presupposed to be 0.9. The body angle is a value between 180 and -180 and the neck angle related to body is a value between -90 and 90.

To define states we need to quantify these three continuous parameters. Uniform clustering is used to quantify them. Using uniform clustering, we supposed that probability is 11 clusters, the body angle is 40 clusters, and the neck angle is 20 clusters. We mean that for example, the rang [-180, 180] for body angle is divided equally into 40 parts (i.e. the first body angle cluster has a body angle value between -180 and -171 and so on).

Thus, in this problem, the number of states is 11*20*40+2 or 8802. The two last states are the target states, i.e. *goal* and *save*. It should be said that the actions are only ″to shoot toward the goal″ and ″not to shoot toward the goal″.

A program for gathering training data and the Q-learning algorithm program were written in C++ language. By applying the training data set on the Q-learning algorithm, a table of Q-values with 8802 entries was resulted. The Q-values, ranging from 0 to 300, indicated that sometimes when the probability, which is calculated by UvA scoring policy, is 1, shooting toward the goal is not the best action; in other words, sometimes when the probability is 1, shooting toward the goal does not guarantee a goal.

Thus, we can say that the body and the neck angle of the goalkeeper are effective parameters, too. The average of Q-values on neck angle related to body angle and the average of Q-values on body angle related to neck angle are represented in figure 7 and figure 8.

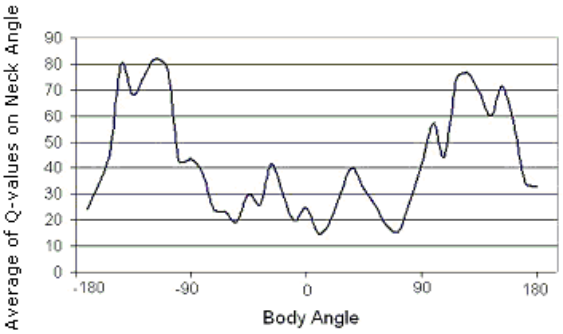


Figure 7: The average of Q-values on neck angle

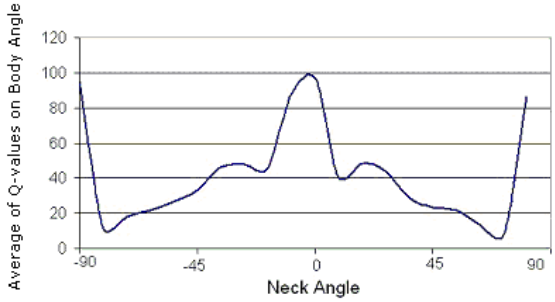


Figure 8: The average of Q-values on body angle

For testing the results of Q-learning, a program is written in which players in each goal position gather all the required parameters, and then obtain related Q-value from the Q-table. If this Q-value is greater than a Q-value threshold, the player shoots toward the goal. To calculate the best Q-value threshold, we compute training error for each Q-value threshold, from 0 to 300. Figure 9 represents the diagram of the training error versus different Q-value thresholds. This figure shows that the Q-value threshold 100 has the minimum training error; thus, we consider the Q-value threshold 100 in this research.

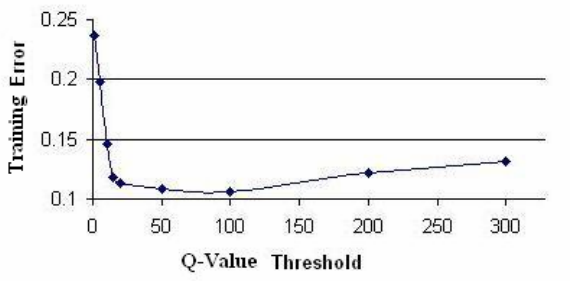
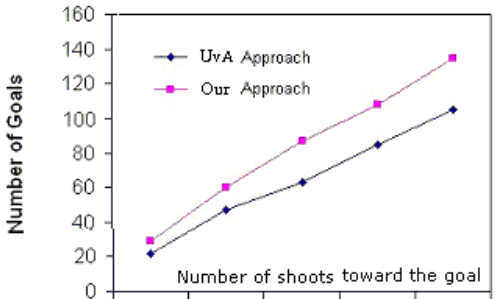


Figure 9: Training error for each Q-value threshold

Figure 10 shows a comparison between UvA approach and proposed approach in this paper. In this diagram, the horizontal axis is the number of shoots

toward the goal and the vertical axis shows the number of goals.



	100	200	300	400	500
UvA Approach	22	47	63	85	105
Our Approach	29	60	87	108	135

Figure 10: Number of goals in UvA approach and our approach

5 Conclusion

In this paper, two parameters, the body angle and the neck angle of the goalkeeper were added to the scoring policy of UvA team. This behavior was trained by reinforcement learning algorithm.

The results show that these parameters affect the result of a shooting toward the goal. Thus we can have better scoring behavior by considering them.

The next step in this research would be using fuzzy techniques [9] for training; the reasons are as follows:

- Large amount of parameters
- Noise and uncertainty
- Large amount of training data which is needed for training
- The problems of the quantization of the continuous values to define states

Another future work would be using a higher number of effective parameters in learning; for example, the number of teammates and adversary players around the goal, or conditions and positions of them.

6 References

[1] Kitano H., Asada M., Kuniyoshi Y., Noda I., Osawa E. & Matsubara H., "RoboCup: A Challenge Problem for AI", *AI Magazine,* 18(1): 73-85, (1997).

[2] Chen M., Foroughi E., Heintz F., Huang Z., Riley P., Wang Y., Yin X., "RoboCup Soccer Server", User Manual, (2003).

[3] Kitano H., Veloso M., Matsubara H., Tambe M., Coradeschi S., Noda I., Stone P., Osawa E. & Asada M., "RoboCup Synthetic Agent Challenge", (1997).

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

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

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

[7] Barto A.G. & Sutton R.S., *Reinforcement Learning: An Introduction*, MIT Press, (1998).

[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] Nguyen H.T. & Walker E.A., *A First Course in: Fuzzy Logic*, CRC Press, (2000).