Abstract. This paper addresses the research was done by Aria2D soccer simulation Team for preparing Robocup 2005. 2D soccer simulation environment provides a distributed, complex, dynamic, and real-time environment. We believe that the problems regarding this test-bed can not be solved with a typical programming in a reasonable way. Therefore, we paid more attention to use machine learning method for solving the various problems of this test-bed.

1 Introduction

Aria soccer simulation team began its activities on Robocup domain in December 2001. In the first attempt, Aria could take 7th place at Robocup international competition 2003 Padova. In the next year competition, because of having some technical problems in the matches, it couldn’t advance from second round. After the competition, Aria team paid more attention to use learning machine methods for solving various soccer simulation problems. Several researches were performed in the machine learning context by other teams. For instance, supervised learning was used in intercept skill and pass evaluation in [1] and reinforcement learning was used in [2,3] successfully. This paper addresses the use of Sarsa Lambda methods for implementing the intercept skill and explaining the experimental results in comparison with analytical one.

2 Learning the intercept skill

One of the most important skills used by soccer simulated agents is intercepting the ball. By using the intercept skill, agent is able to get the moving ball at any distance. This skill affect on offensive and defensive affairs. It used to get the ball from the opponents, receive pass from teammates, and catch the free ball. At least, in any moments, one of the players of each team is performing this skill. Main goal of intercept skill is determining optimal point to reach the ball based on current location and velocity of the player and the ball, and then going to that point with maximum speed. Because of noise existence in ball displacement and not having complete information from environment, it’s not simple to implement this skill. Existence of noise causes
that anticipating next locations of the ball and calculating optimal position for catching
the ball become complicated. Suppose that the player recognizes the receiving
location of the ball. Then he moves toward this point. If during this movement, for
mentioned reasons, the location of the receiving ball is changed, it should have a turn
toward this point. This turn considerably increases the time of reaching the ball.

Various methods are proposed and used to implement this skill. For example, in a
method which is offered in [4], first possible point for reaching the ball is calculated
by predicting the ball position in next cycles. Other empirical methods such as supervised learning with using neural networks [1,5], reinforcement learning with real-time
dynamic programming [2], and learning automata are used to implement this skill. In
new research used in Aria team, learning of this skill has been done with Sarsa Lambda using linear approximator with tile coding and due to cognition of operation
of learner method; it is compared with analytical methods.

3 Reinforcement Learning

Standard episodic reinforcement learning is a framework for interaction between
learner agent and Markov’s decision making process. Each episode contains T time
steps, \(s_0, a_0, r_1, s_1, a_1, ..., r_T, a_T, s_T\) with the states \(s_t \in S\), the actions \(a_t \in A(s_t)\),
the rewards \(r_{t+1} \in R\), which are the random variable with the mean \(\mu_{s_t}\), and the next
state \(s_{t+1}\) is chosen \(p_{s_{t+1}}^{a_t}\).

Given a state, \(s_t\), \(0 < t < T\), the action \(a_t\) is selected according to probability
\(\pi(s_t, a_t)\) or \(b(s_t, a_t)\) depending on whether policy \(\pi\) or \(b\) is in force. We always use
\(\pi\) to denote target policy, the policy that we are learning about. In the on-policy
case, \(\pi\) is also used to generate the actions of the episode. In the off-policy case, the
actions are instead generated by \(b\), which we call the behavior policy. Sarsa Lambda
is an on-policy method the details of which can be found in [6].

4 Learning Scenario

For generating the intercept examples, we define a training scenario. The training
scenario is:
- Coach places the ball at a specific position with arbitrary initial velocity.
- Coach places the player at a random position around the ball and announce the
  start of episode.
- Player selects an action in each cycle and gets the previous action’s rewards
  until it reaches the ball.
- When the player reaches the ball, coach announces the end of the episode and
  the next episode is started.
Notice that the agent should consider location of the ball and itself in a relative co-
ordination. With relative coordination, the learned skill can be used in the other part
of the field. Fig. 1 shows an example of initial situation of an episode.

Fig. 1. Initial situation of an episode

5 Experimental Results

To be able to evaluate the learning method, we need a parameter which shows how
the player is successful in an episode. We use the number of turns which are issued
by the agent in the episode as our performance measurement. Small value of the per-
formance parameter leads to better intercepting the ball. With the scenario mentioned
in section 4, episodes are generated. In each episode, agent selects next action using
the Sarsa Lambda method and also updates its learning weights. With passing the
time, agent’s performance is increasing more and more and learning method eventu-
ally converges to intercept the ball with a minimum number of turns. In the Fig. 2, the
mean of the number of turns over episodes is shown. In the beginning of the learn-
ing, agent just does exploratory actions, so the performance measure is about 17 turns
per episode. After finishing the 32000 episodes, the mean of the performance param-
ter becomes less than 5. When the learning is over, we repeat the experiment with
learned skill, but now the agent does not perform the exploratory actions. In this case,
the mean of the number of turns becomes 3.10. If we do the experiment with analyti-
cal method which was used in Aria2004 team, the performance parameter becomes about 4.08. In table 1, summarized results are shown.

![Graph showing learning the intercepting skill](image)

**Fig. 2.** The figure shows the performance of the agent during episodes.

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean of the number of turns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sarsa Lambda</td>
<td>3.10</td>
</tr>
<tr>
<td>Analytical Method</td>
<td>4.08</td>
</tr>
</tbody>
</table>

**Table 1.** The mean of the number of turns per episode.

---

6 Conclusions

This paper describes the learning of intercepting skill with Sarsa Lambda algorithm and also presents the experimental results in comparison with analytical one. Our experiment shows that learning method can perform much better than analytical intercepting method used in Aria2004 source code.

We suggest using the model-based methods in future works, because Sarsa Lambda is model-free algorithm, but soccer agents have a model of environment. This model is not complete and is noisy one. Even in this case, we think using the model-base learning method may converge faster and better than model-free methods.

References