The New solution to Cooperation Strategy in Dribbling Task of RoboCup

Zhang Wei-bing[1,a], Liang Chu-guang [1,b], Li Gang-cheng[1,c]

[1]Dept. of Information Engineering Hunan College of Information & Engineering Changsha, China

Changsha, China

[a]Zhangweibing@mail.hniu.cn, [b]Liangchuguang@mail.hniu.cn, [c]Ligangcheng@mail.hniu.cn

Keywords: Reinforcement learning; Dribbling; Possession algorithm

Abstract. This paper proposes a solution based on reinforcement learning in the soccer dribbling tasks of RoboCup. The dribbler spares no efforts to keep possession of the ball from beginning to end while adversaries attempt to gain possession. In this paper, the dribbler uses SARSA possession algorithm and the opposite perform traditional strategies respectively. These two kinds of strategies are applied to robots in the environment of 4V3.The results show that by learning many times robots have made great progress in terms of dribbling time.

Introduction

RoboCup simulation game provides a fully distributed control and real-time asynchronous multi-agent environment. Through this platform, you can test various theories, algorithms as well as Agent architecture. And you can also research cooperation and confrontation problems between multi-agent in real-time asynchronous and noisy confrontation environment. The game is conducted using the Client / Server mode in a standard computer environment when teams write their own client program to simulate the real soccer players to compete.

Applying reinforcement learning algorithm to robot soccer simulation in RoboCup must overcome several problems such as multi-dimensional continuous state space, the effect of noise, the multi-agent and the need for real-time action. In the past period of time, machine learning has been applied in many RoboCup sub-tasks. This paper presents the SARSA algorithm in multi-agent (4V3) dribbling task and desired results have been achieved after a period of study.

Reinforcement learning

The theoretical foundation of reinforcement learning

Reinforcement learning, also known as stimulate learning or assessment learning, is a process during which agents constantly test and learn to make the value of the accumulated reward of system behavior to get the maximum from the environment. The basic model is shown in Figure 1. In reinforcement learning, we design algorithms to put the external environment into actions to maximize the amount of reward way. The agent has not been told what to do or what action to take, but discoveries by looking at which action been most rewarding. Agent movements affect not only immediate rewards, but also the next action and the final reward.

In reinforcement learning, the environment is in a certain state s of the state set S and the agent selects an action a in action set A. The action a acting on the environment will immediately receive a reward r, when r is greater than 0 the agent will enhance the tendency of this action in the future. When the control behavior of learning system repeatedly interacts with the state and evaluation of the environment, mapping strategies from state to action have been constantly modified by the way of learning in order to optimize system performance.

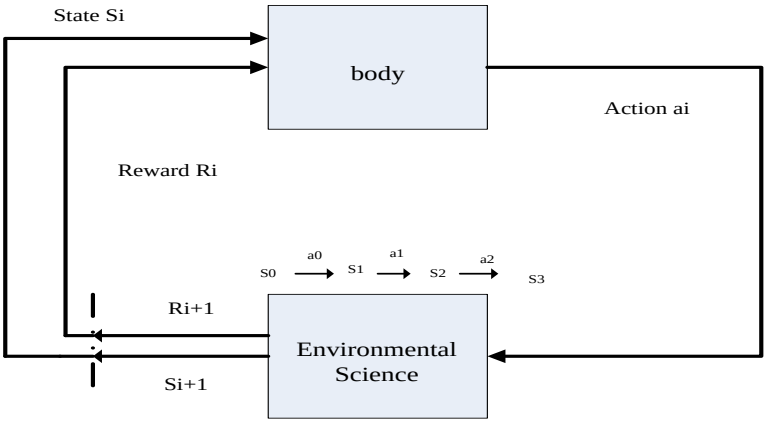

Fig 1 model of reinforcement learning

Reinforcement learning includes the four following main elements:

（a） Strategy: it defines agent’s behavior way at a given time, directly determines the agent's actions and is the core of reinforcement learning. Good or bad strategies directly determine the overall performance of agent behaviors.

（b） Reward function: it defines goals of reinforcement learning, maps perceived environmental conditions to a reward signal r and evaluate the produced action is good or bad. Reward function is often determined objectively and can be a standard of changing strategy.

（c） State value function: it is a long-term perspective to determine the current action is good or bad, where value function Q represents expectation of sum of reward discounts when the action a and follow-up strategies are carried out in the state s, denoted $Q(s,a)$.

（d） Environment model: it is an optional component to certain reinforcement learning systems. Environmental model is simulating the behavior of the environment. Using the environment model, the agent can consider possible future scenarios but not actually experience to make planning at the same time they make decisions.

Implementation of reinforcement learning algorithm

In this paper, SARSA learning algorithm has been introduced, the specific process is as follows:

（a） initialize all value functions Q(S,A)

（b） repeat every step following

initialize state s

repeat (every step of the stage)

choose an action a according to value Q of the current state（e.g.,ε-greedy）

perform the action a then observe r and s'

choose action a' of next time according to s' and selected strategy（e.g.,ε-greedy）

update Q(S,A), the updating rule is $Q(s,a)=Q(s,a)+\alpha[r+\gamma Q(s',a')-Q(s,a)]$

s←s',a←a'

until s is the target state

Application of reinforcement learning in dribbling task of RoboCup

In Robocop, dribbling is a significant sub-task. Because to some degree who controls the soccer is in control of the initiative of the game. In the specific process of the game, the dribbler is bound to do everything in control of the ball, not letting it fall into the hands of adversaries, while the rivals exert all their skills such as taking sliding tackle and so on to snatch the ball[2].Applying reinforcement learning algorithm to possession strategies makes players do learning in the process of keeping possession of the ball. After a period of study, it’s excellent for agents to take the best action to get greater grades in the game.

In the RoboCup robot soccer simulation platform environment is a continuous state, in this experiment we use some state variables to represent the state. What differs from methods designed for single-agent status is in methods designed for multi-agent status ,it is essential to take account of

the perspective and the distance between own party and the defense. In the environment of 4V3, as is shown in figure2, supposing that k1 possess the ball, at this point a certain status is composed by 19 state variables: dist(k1,C); dist(k1,k2); dist(k1,k3); dist(k1,k4); dist(k1,t1); dist(k1,t2);dist(k1,t3);Min(ang(k2,k1,t1),ang(k2,k1,t2),ang(k2,k1,t3));Min(ang(k3,k1,t1),ang(k3,k1,t2),ang(k3,k1,t3));Min(ang(k4,k1,t1),ang(k4,k1,t2),ang(k4,k1,t3);dist(k2,C);dist(k3,C);dist(k4,C);dist(t1,C);dist(t2,C);dist(t3,C);Min(dist(k2,t1),dist(k2,t2),dist(k2,t3));Min(dist(k3,t1),dist(k3,t2),dist(k3,t3));Min(dist(k4,t1),dist(k4,t2),dist(k4,t3)). (where dist(a, b)stands for the distance between a and b, ang(a ,b ,c) represents the size of angle whose vertex is b, consisting of three points a, b and c. And C is the center point of the learning area.)

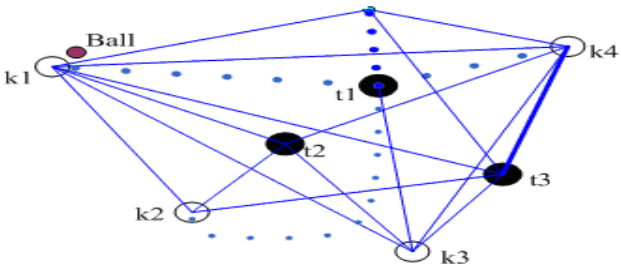

Fig 2 setting the state variables in the environment of 4V3

In the action designs, there have been four actions designed including holding ball by his own, passing to the nearest player, passing to the second nearest player and passing to the third nearest player. When the action is selected, what based on is the ε-greedy produces the maximum value of Q selecting all the actions in the current state.

When it comes to reward function, the reward is determined by the time of robot's possession of ball in the dribbling task of RoboCup. Put it another words, the longer the robot possesses ball, the greater reward he gets. The whole task of the learning is to maximize robot's value of the accumulated reward, that is to maximize the robot's time of possession of ball[1]. Specifically you can use the following formula (1):

(1) $\text{Reward}=T_{\text{the current action time}}-T_{\text{the last action time}}$

Empirical results and analysis

In this experiment, we compare the robot's possession time of traditional non-enhanced learning strategies with those of reinforcement learning (no communication between robots) and reinforcement learning (there is communication between the robots). The experimental data is obtained at $\varepsilon = 0.01$, $\alpha = 0.15$, $\gamma = 0.9$ when using MATLAB programming and experimental graphics are shown in Figure 3:

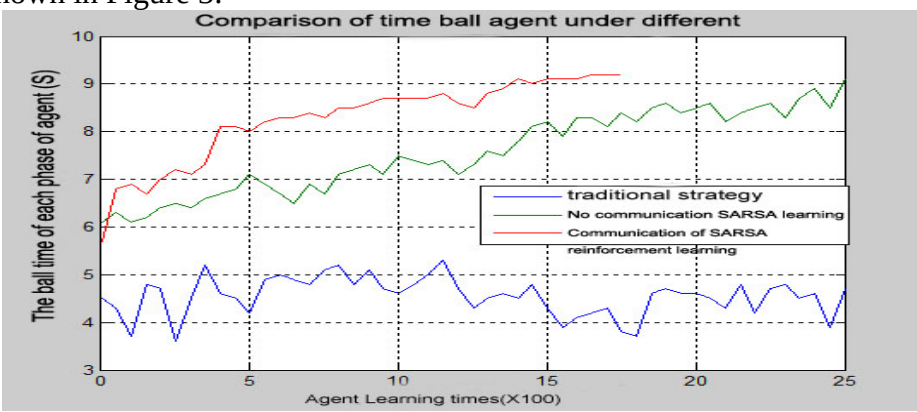

Fig 3 agent possession using different strategies

As is indicated above, it is obvious that the time of possession of the ball by using the reinforcement learning algorithm is significantly better than that by performing the traditional strategy, which suggests that after a period of study, the team conducting SARSA algorithm strategy will be in active position in the game, then the team can choose to be offensive or defensive

according to the situation of the court. What's more, learning efficiency of communication SARSA algorithm is significantly greater than that of non-communication SARSA algorithm. And the dribbling time by using communication SARSA algorithm tends to converge quickly, in theory (training time is infinite) the dribbling time by using non-communication reinforcement learning is almost equal to that by carrying out communication reinforcement learning.

Conclusion

Reinforcement learning has made great success in the dribbling sub-tasks of multi-agents. But with the number of agents increasing, the size of the state space shows growth in the way of index. So how to apply the reinforcement learning algorithm to more agents to cooperate with others still need to be resolved.

The innovation in this paper lies in that SARSA algorithm is applied to robot soccer simulation game in dribbling sub-tasks, and great success has been achieved in the dribbling task of agents, which allows players to increase the dribbling time, take control of the initiative in the game and win at the end of the game more easily.

References

[1] Matthew E.Taylor,Peter Stone,Yaxin Liu. Transfer Learning via Inter-task Mappings for Temporal Difference Learning[C].Journal of Machine Learning Research8(2007):2125-2167

[2] Arthur Carvalho and Renato Oliveira.Reinforcement learning for the Soccer Dribbling Task[J].Proceedings of 2011 IEEE Coference on Computational Intelligence and Games.2011

[3] T.Gabel,M.Riedmiller,and F.Trost,"A Case Study on Improving Defense Behavior in Soccer Simulation 2D:The neuroHassle Approach,"[C].in RoboCup-2008:Robot Soccer World Cup XII,2009:61-72

[4] Wei Li.Reinforcement Learning in Keepaway Framework for RoboCup Simulation League[J]August,2011

[5] Robin Soetens.Reinforcement Learning applied to Keepaway[D],Master thesis from Eindhoven University,2010

[6] Peter Stone,Richard S.Sutton,and Satinder Singh[J].Reinforcement Learning for3VS.2 Keepaway.In RoboCup-2003:Robot Soccer World Cup VII

[7] Matthew E.Taylor,Peter Stone.Transfer Learning for Reinforcement Learning Domains:A Survey[C].Journal of Machine Learning Research10(2009):1633-1685