

HfutEngine2008 Simulation 2D Team Description Paper

Long Li, Hao Wang, Dunqiao Bao, Baofu Fang, Jingjing Jiang, Li Shang

Artificial Intelligence And Data Mining Lab

Abstract. This paper describes the background, the framework and the design feature of the HfutEngine2008. We put forward our decision-making model based on max-value, and narrate the components of the decision-making model in details. We hope more people could understand the HfutEngine2008.

1 Introduction

Team HfutEngine was founded in 2002 and took part in the RoboCup2002 of China. In the following years, HfutEngine is developing fast and joining many matches. In RoboCup China Open2007 we took the 2nd place of simulation soccer 2D. We hope to obtain a better grade in World RoboCup 2008. It is also the first time for us to take part in The World RoboCup. We want to probe into Multi-Agent System and robocup with everybody who is interested in them.

2 Framework of HfutEngine2D

In our exploitation we found that any Multi-Agent cooperation was based on how the single agent adapt to the Multi-Agent System. If every element in the system can accommodate the system, the system is steady. It is not need to have unitive command for every agent. Our strategy is based on the value judgment, that is every Agent has its own evaluator to calculate correspond value. Then the action which have max value is being executed by executant.

This framework first founded in 2005. The former methods make use of evaluator to design to do something like shoot, dribble, etc and then to perform the relevant action. In this way, the decision of the evaluator mainly depend on experience knowledge. Now we depend on the product of income and the probability of success which are estimated by environment beforehand to decide the action. The idea is not easy to perform. The player decide to perform an action by the balance of the value.

This structure is shown in Figure 1.

3 The Highness of HfutEngine2008

The HfutEngine2008's high decision includes two parts, evaluator and executant. Evaluator predict the benefits token by all the actions, then we can Make a

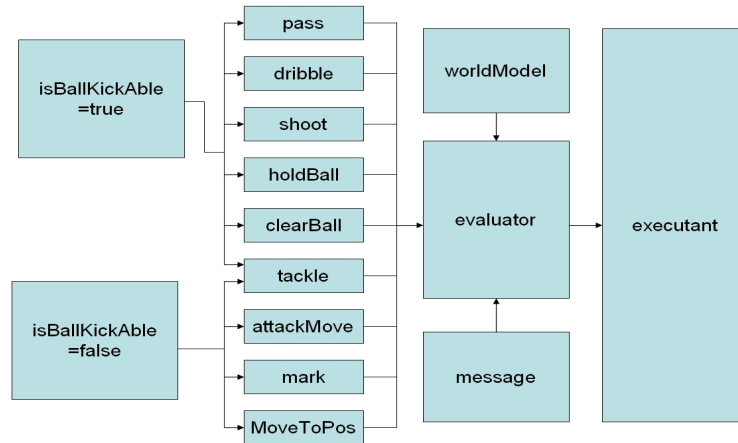


Fig. 1. The structure of HfutEngine2D.

decision which action should be executed. Executant take responsible for how to execute the action.

Executant includes tackle, holdBall, clearBall, dribble, shoot, pass, attackMove, mark and MoveToPos. First, if the ball is kickable , the value of the action is set to zero except five actions :tackle, clearBall, holdBall, dribble, shoot, pass. Whereas these five actions(except tackle) are set to zero if the ball is not kickable. When the ball is kickable we transfer the function pass included by executant to decide how to pass (not really execute the action, just create the command,not put into the command queue), then multiply the success percentage of the action with the value of benefit brought by the action. As a result, we can get all kinds of the values from the different passing paths.Choose the best one (whose value is maximal) and compare with other values produced by other actions.Finally, we get a max value of the collectivity actions, then execute the action(put the action into the command queue).

3.1 Evaluator

How to get the best value. Take the pass ball for example. Just as the thought we refer above, we get the value of pass by the product of the success percentage of the action pass and the value of benefit brought by the pass.The success percentage of the action pass is obtained by large numbers of pass-trainings.Set a initial scene, including four parameters, they are the velocity of ball (ballVel), the distance to the Player who is intercepting the ball(dist), and this player's body-angle(angBody) and velocity(agentVel). Through the result of training we can get the success probability of the interception in different time.

Suppose occur A and B as followings:

A : a teammate intercept the ball within the time t .

B : the fastest opponent do not intercept the ball within the time t .

$P(A)$: the probability of the event A occur.

What should us do is to find a time t to make the expression $|P(A) - 0.8|$ achieve min value. Thus the $P(A|B)$ is the success percentage of the interception. The benefit is also obtained by value evaluation. The value is computed by the expression below

$$value = posValue + \sum_{i=1}^n x_i \cdot w_i.$$

In the expression, posValue is the adjusting of position . every position have a fixed value(posValue), which gained by lots of training. For example, when we shoot successful, we write down the path of the ball. Every point of the path represent a Value.

We can see the posValue from figure 2, the deeper the color of the point is, the bigger the Value is.

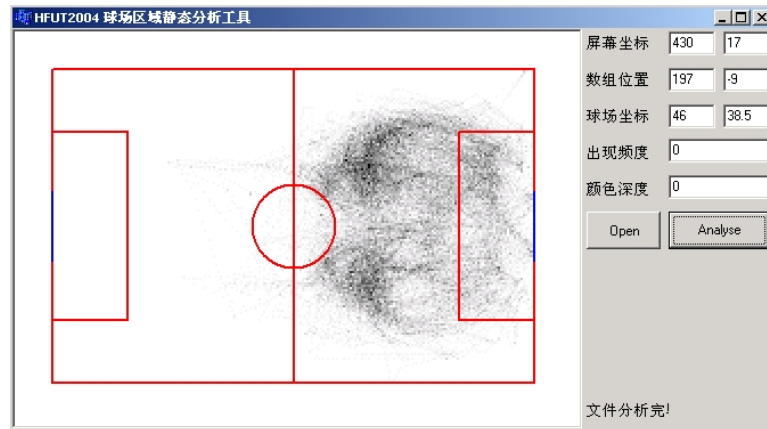


Fig. 2. The value of different position

In the expression, x represent true or false of a attribute. For example, if is-Shoot-Able is true, then x equals 1, else x equals 0. There are many much attribute like this. eg: whether the players break through the defender or not, is there opponents in the interception path etc. In the expression, w is the weight of attribute. Different attribute has different weight. The attribute which can have most effect on the score, the weight is the biggest, on the contrary, the weight is smallest. All of the value of the weight is obtained by lots of training and conclusion .

3.2 Executant

The executant includes how the player execute the actions. Some of them are gained by training, others are deduced by the mathematic and physical expressions combined with the rules of the server. Take shoot for example.

The action shoot is obtained by training. The training scene can be seen at figure 3. The arithmetic is as follows:

(1)Fix the position of the ball and the shooting-player. Set the position of the goalie randomly and record the position.

(2)If the shooting-player take the ball, then shoot to the fixed position in the goal.

(3)If the ball is kicked in the goal, then record the shoot is success ,whereas if the ball is hold up by the goalie then record the shoot is failure.

Repeat the (1) to (3),we can get a set of value of posGoalie , base on the value of the ' posBall 'and ' posShootingPoint ', then we can compute the α and d . We use network of Radial Basis Function to train these values. Make the Gauss Nucleus Function and network framework as two inputs and one output. We have 3 parameters to learn including the center of Radial Basis Function, the weight from center-cell to output-cell, variance. The training result show that the ratio of success is up to 91.28Shoot-Advisor need to submit Method to the higher framework. The method need 3 parameters.

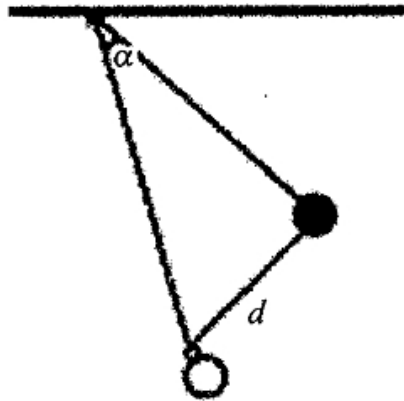


Fig. 3. Training for shoot

4 Other Features

4.1 Roles-reversed

We make the roles reversed by a protocol which like the two-handle protocol. The Agent who want to reverse roles send the role-reversed order to other players by say-order and to attention to the requested-player. If a player receive the recent request and the requested-player is not him, he will send it to the requested-player. When the requested-player receive this request, if he accept this proposal, he will send a agree-order. Then the action of the role-reversed will be executed.

We can see it in Figure 4.



Fig. 4. Roles-reversed

The Number 3 is standing in the defensive area of the number 2 and the number 2 is nearer to the the point of number 3. At this time if the number 3 return to its own area, he would waste time and stamina .So the number 3 send the role-reversed request to the number 2 and hope the number 2 to defend the opponents. The number 2 hear the request and accept it. So the two player finish the role-reversed.

4.2 Reverse Design

Although we think the design idea of value-judge is a good idea, it create bug sometimes. Thus we make another way called Reverse Design to help the value-judge framework to make the team work better. First we make the shooting as

the first step because it is most probably to get score. Then make the pass or the dribble as the second step because it is most near to the shooting-condition that is make sure whether the pass-then-shoot and the dribble-then-shoot is true or not. Make the same method. When a player come into one of steps, he will make decision based on the tactics across the value-judge framework. We see that: In the figure 5, number 11 come into the third step and he work to a receive point. In figure 6, the condition of pass-then-shoot of number 10 is true, so it come into the second step and pass the ball to number 11. In figure 7, the shoot-condition of number 11 is true, it come into the first step and shoot.



Fig. 5.



Fig. 6.



Fig. 7.

5 Conclusion and Future Works

The practice prove that the design idea of value-judge is very successful. The achievement that we have got in the past year further improve that point and the ability of our team make a great leap. In Table 1, we can see the result of competing with some teams. It shows that HfutEngine2008 with this design have good match-ability.

Table 1. the result of competing with some teams

Team	Ave Goals Scored	Ave Goals Conceded	win	draw	lose
HfutEngine2005	8.5	0.2	20	0	0
HfutEngine2006	4.9	0.15	20	0	0
HfutEngine2007	2.1	0.55	13	4	3
Mersad2005	2.25	0.7	15	3	2

We hope that we can improve more segments of our team. We will further optimize this framework and exert its flexibility advantage. We will solve the problem of our team gradually. Also, we plan to make a new way to forecast the situation in the playground based on experience which include the states of teammates, opponents and ball and so on. Based on this new way we can get the value more correctly. We also continue to make research on the Multi-Agent System and Machine Learning in order to enlarge the ratio of learning in our team. Meanwhile, the research will be focus on the fast-online learning not the off-line accumulated learning. The fast-online learning make the player learn to change on-time. In the coming time we will work hard to make a good result in the World RoboCup.

References

1. Zhulin AN, Jingjing YU, Hao WANG. Robocup Simulation League Goalie Design. Proceedings of 1 Austria Open of RoboCup. 2003.
2. Xiaoping Chen, et al, Challenges in Research on Autonomous Robots, Communications of CCF, Vol. 3, No. 12, Dec 2007.
3. Kitano H, Tambe M, Stion P, et al. The Robocup synthetic agent challenge 97[A]. Proceedings of Fifteenth International Joint Conference on Artificial Intelligence (IJCA-97)[C]. Berlin:Springer,1997.24-29.
4. Baofu Fang, Hao Wang, et al. The Survey of HfutEngine2005 Robot Simulation Soccer Team Design, Journal of Hefei University of Technology. Vol.29(9). Sep, 2006.
5. Gao Jian-qing, Wang Hao, Yu Lei. A Fuzzy Reinforcement Learning Algorithm and Application in RoboCup Environment. Computer Engineering and Applications, 2006(6).
6. Yang LIU, Hao WANG, Baofu FANG, Hongliang Yao. Application of the method of support vector regression in RoboCup. Journal of Hefei University of Technology Vol.30(10). Oct,2007.