YuShan2022 Team Description Paper for RoboCup2022

Zekai Cheng,Jinbo Guo,Yahui Ren,Yaokun Tang,Luyi Rong
Department of Computer Science,
AnHui University Of Technology, MaAnShan, P.R. China
Yushan2d@gmail.com

abstract: This team description paper describes the team optimization direction and method of Team YuShan2022 this year. The team strengthens the team's offensive ability from the direction of offensive movement. The offline training model is adopted, the model is run online, and the LambdaMART algorithm is used to add state information for the selection of the running position, and correct the local position. The experiment proves that the new running method improves the scoring ability.

1 Introduction

Team YuShan is affiliated to Anhui University of Technology and has participated in seven Robot World Cup competitions since 2012, including fourth place in the 2019 World Cup in Sydney, Australia, and third place in the 2021 World Cup in France. Team YuShan participated in the RoboCup China competition and won the "three consecutive championships" in 2016-2018, the runner-up in the 2019 RoboCup China competition, and "two consecutive championships" in the 2020-2021 RoboCup China competition. In recent years, the YuShan team has used data mining technology to analyze the characteristics of the team, and on this basis proposed a digital twin framework. It has achieved certain results in the aspects of formation, player movement, passing analysis, shooting strategy and judgment of offensive and defensive state. YuShan2022 is developed based on YuShan-Base, and YuShan-Base is developed based on Agent-2D 3.1.0 [1] version.

2 Improved frame of positioning method

An important problem in robotics is to determine the actions of the agent based on the situation and goals [2]. In the offensive state, how to use the court state, design flexible tactical team strategies and excellent action selection is very important and difficult . In Agent2d, the game formation is realized by Triangulation based Positioning Mechanism[3] (TBPM). The main idea of this model is situation based strategy positioning (SBSP) [4]，The target points of movement are designed with the help of formal judgment, tactics and the distinction of player roles, using human experience and machine learning algorithms. But after editing the formation file, it becomes a fixed formation. Usually based on the position of the ball, the target point of the player's movement is obtained. For every position of the ball on the court, the player will have a fixed and unique position corresponding to it. It can ensure that the overall trend and formation of the team is complete, but it is not enough to meet the needs of various complex situations.

As shown in Figure 1, based on the idea of data enhancement, we enhance the original target point that is uniquely determined according to the position of the ball, take the target point as the center and 1 meter as the unit to generate a 9×9 grid. The vertices of each mesh are candidate target points, and new target points are selected from the candidate target points.

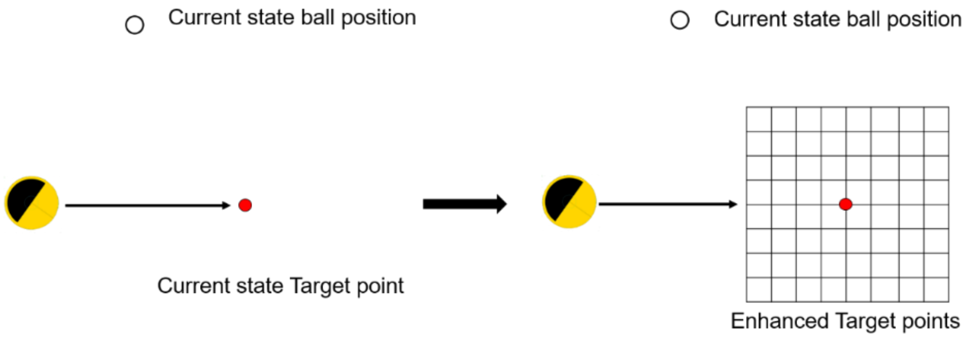
○ Current state ball position
○ Current state ball position
Current state Target point
Enhanced Target points

Figure 1 Target point enhancement

As shown in Figure 2, we improve on the original offensive strategy movement framework. The selection of candidate target points is transformed into the selection of candidate states. Combine the generated candidate target points with the current state, take the candidate target points as the position of the runner, and generate multiple candidate states under the condition that only the position of the runner is changed. Use the sorting module to select the pros and cons of the candidate states, and select the best candidate state. The candidate target point corresponding to the best candidate state is the new moving target point. Each new target point is a correction to the original target point. In this way, the overall trend and formation of the original system can be preserved and the basic performance can be guaranteed. It can also avoid the problem that the algorithm is difficult to converge because the state space is too large. It can also reduce the module's excessive dependence on the algorithm, alleviate the complex and difficult to solve problems such as the large difference between the theoretical and experimental results caused by the systematic error accumulation of the algorithm, and increase the robustness of the module. At the same time, more state information is added to the influencing factors that affect movement, which makes players more flexible and can optimize the movement of offensive strategies.

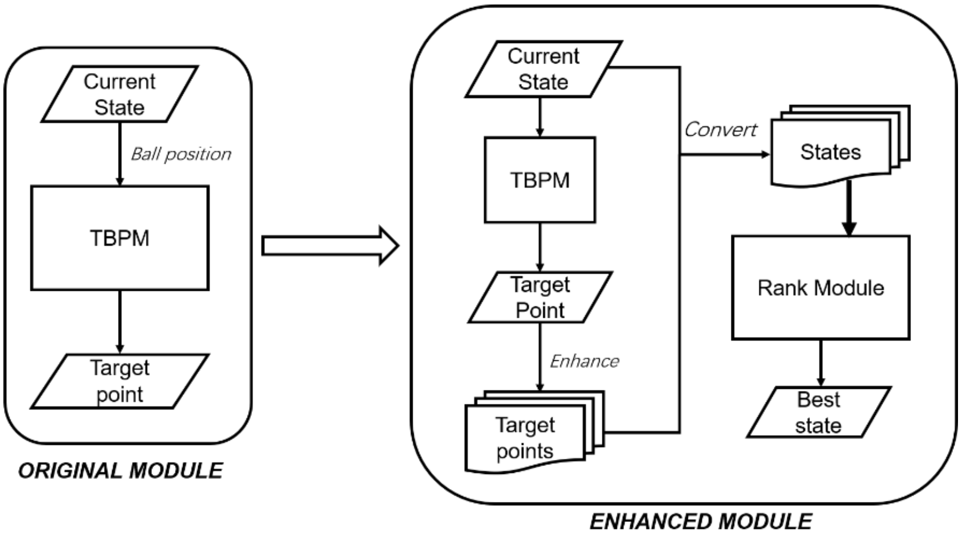

Figure 2 Improved positioning module

3 Offensive move realization

As shown in Figure 3, we use offline training and online application to achieve offensive movement. We mark the state information according to the degree of correlation between the results caused by the state information and the theme of the offensive strategy in this state to form training data. The characteristics of the state data are processed according to the simulation rules, in line with the mode of the player's perception information as much as possible, and the state that is most in line with the current offensive strategy is selected from the numerous inference state information through a sorting learning algorithm, and the player's movement is used to approach the best state. , to create an enabling environment to optimize the performance of the simulated team.

We apply the LambdaMART [5] algorithm to solve the candidate state ranking problem. The LambdaMART algorithm is a ranking learning algorithm, and the ranking model is designed to rank, i.e., to generate a permutation of items in a new, unseen list in a similar way to the ranking in the training data, a supervised learning process. The LambdaMART algorithm incorporates ranking indicators into the learning process, which is a pairwise method, and we use Normalized Discounted Cumulative Gain (NDCG) [6] as the ranking indicator. Because the NDCG can not only deal with multiple related levels, but also give the degree of correlation between the state and the theme of the offensive strategy. In addition, LambdaMART pays more attention to the ranking of the top of the sequence than other sorting algorithms, which is consistent with our requirement to select the best candidate state from 81 candidate states.

The result of the LambdaMART model is composed of many decision trees through the Boosting idea. The fitting target of each tree is the gradient of the loss function. The gradient is calculated by the Lambda method. Lambda is defined as follows:

$$\lambda_{ij} = \frac{\partial C(s_i - s_j)}{\partial s_i} = \frac{-\sigma}{1 + e^{\sigma(s_i - s_i)}} |\Delta NDCG|$$

In addition, LambdaMART pays more attention to the ranking of the top of the sequence than other sorting algorithms, which is in line with our need to select the best candidate state from 81 candidate states.

We use the offline trained model for online operation, and then use the output log file for training, looping continuously, trying to let the agent use a large number of demonstrations to imitate the existing games that have appeared in the past. Favorable states, thereby increasing the frequency of favorable states and tapping the potential of the existing system. We process the log files of the game, select the appropriate target state to form the experience pool, train the learning-to-rank algorithm model, and use it for the selection of the running position when the team moves in the online offensive strategy.

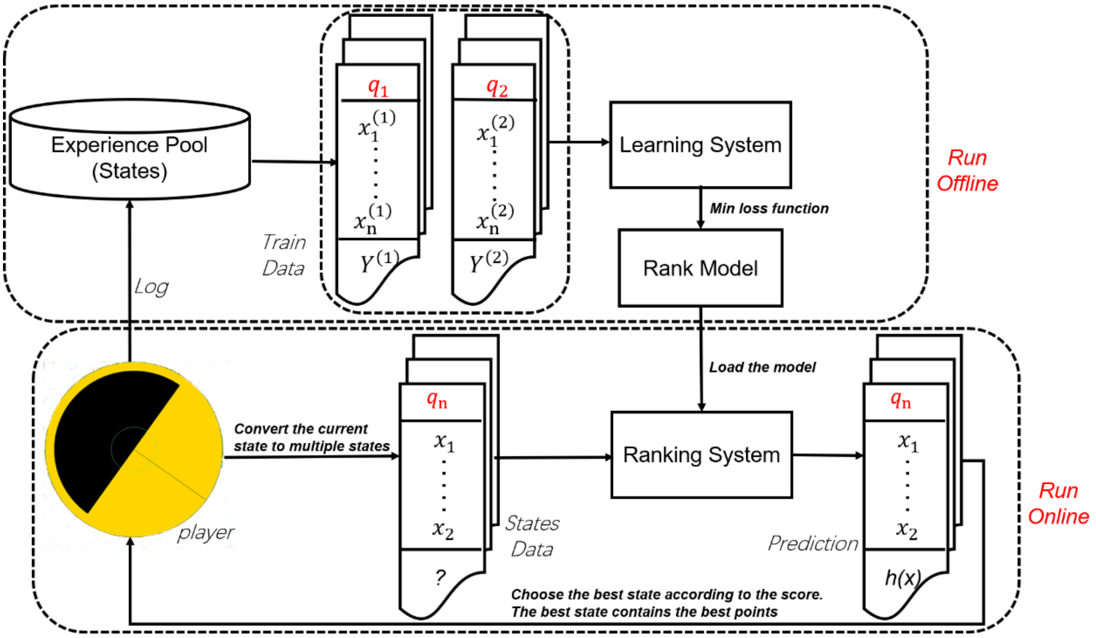

Figure 3 offensive strategy positioning scheme for target point selection

4 Experiments and Results

In the attacking state, the main goal of the attacking strategy is to keep the ball in the attacking state and to score goals for the purpose of winning the game. We use this to divide offensive strategies into conservative and aggressive offensive strategies. Under the conservative offensive strategy, the main purpose of the running player is to receive and pass the ball, cooperate with teammates to control the ball as much as possible in their own side, and at the same time impose a certain offensive threat to find offensive opportunities. Under the aggressive attacking strategy, the running players aim to score goals, interspersed with positions that pose a high threat to the opponent's goal and prepare to shoot. At the same time, this position should be a good receiving point.

We train two ranking models by combining aggressive attacking state data and conservative attacking state data respectively, which are used in the online attacking movement module. We add these two modules to the original code for online testing without changing other modules. The actual combat effect is shown below, in which the point pointed to by the red line is the movement target point calculated by the player in the online movement. The black line is the trajectory of the ball in the later period of time

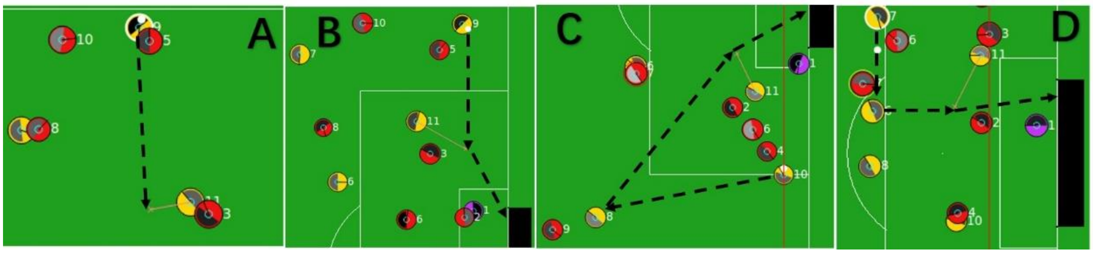

Figure 4 Effect demo

1. Figure A, in a conservative attacking state, when the ball-handling player is threatened, the running player runs to the receiving point to help control the ball
2. Figure B, in an aggressive attacking state, guide the pass, and shoot after receiving the pass
3. Figure C and Figure D, in the aggressive attack state, guide the return pass, and shoot after receiving the pass.

Our ranking model is composed of multiple decision trees through the Boosting idea. Each tree is fitted with the residual of the previous tree fitting, and finally the weak model is superimposed to gradually approach the real situation to form a strong model. We use the XGBoost [7] framework to output the importance of state features, as shown in Figure 5 below.

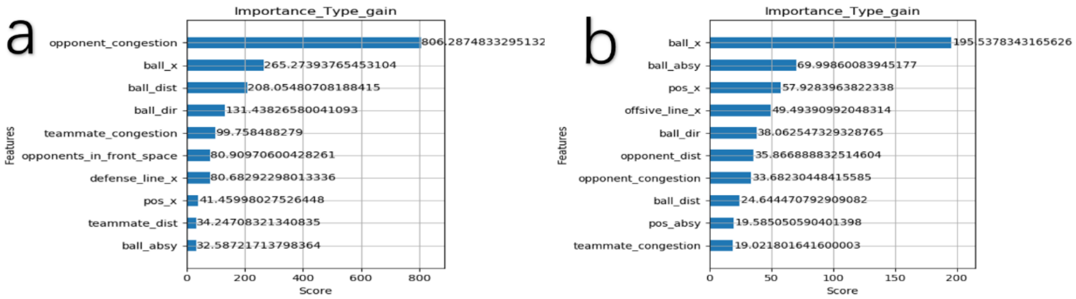

Figure 5 Feature Importance

Gain represents the average gain for reducing the loss function when the feature is used as a tree model division variable. Among them, Figure 5(a) is the ranking model in the conservative state, and Figure 5(b) is the ranking model in the aggressive state. It can be seen from the figure that when the main attacking goal is to control the ball, the model is more sensitive to the crowding information of the enemy players. When the main attacking target is a goal, the x-coordinate of the ball (the larger the x-coordinate, the closer to the opponent's goal) is more important for the model to make judgments。

5 Conclusion

Attack strategies have become a hot topic in the field of multi-agent cooperation. The team's offensive strategy not only depends on the action choices of the ball-handling players, but also the strategic movements of the non-ball-handling players can also play an important role. In real-time, asynchronous, and noisy environments, good positioning creates a favorable offensive environment for the ball handler and can guide the ball handler's decision-making. YuShan2022 improves the offensive strategy movement from two directions of strategy division and target point selection, and proposes a new movement point selection model based on the LambdaRamk algorithm, which re-selects target points around the original movement target points to adapt to the complex Uncertain course environment. The model has been implemented in YuShanBase, and our future work will focus on the transition of offensive and defensive states, and the optimization of action sequence selection to improve the overall performance of the team. I would like to express my sincere thanks to Hidehisa Akiyama, Mikhail Prokopenko, Nader Zare and others for their promotion of 2D Alliance over the years.

References

1. Akiyama, H.: agent2d-3.1.0 RoboCup tools. https://osdn.net/projects/rctools/downloads/51943/agent2d-3.1.0.tar.gz/
2. Nguyen Q D, Prokopenko M. Structure-Preserving Imitation Learning With Delayed Reward: An Evaluation Within the RoboCup Soccer 2D Simulation Environment[J]. Frontiers in Robotics and AI, 2020, 7.
3. Akiyama H, Shimora H, Noda I. Helios2009 team description[C]//12th RoboCup International Symposium.(July 2008)(CD Supplement). 2009: 25.
4. Reis L P, Lau N, Oliveira E C. Situation based strategic positioning for coordinating a team of homogeneous agents[C]//Workshop on Balancing Reactivity and Social Deliberation in Multi-Agent Systems. Springer, Berlin, Heidelberg, 2000: 175-197.
5. Burges C J C. From ranknet to lambdarank to lambdamart: An overview[J]. Learning, 2010, 11(23-581): 81.
6. Jarvelin K, Kekalainen J. Cumulated gain-based evaluation of IR techniques[J]. ACM Transactions on Information Systems (TOIS), 2002, 20(4): 422-446.
7. Chen T, Guestrin C. Xgboost: A scalable tree boosting system[C]//Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016: 785-794.