HELIOS2017: Team Description Paper

Hidehisa Akiyama[1], Tomoharu Nakashima[2], Sho Tanaka[2], and Takuya Fukushima[2]

[1]Fukuoka University, Fukuoka, Japan
akym@fukuoka-u.ac.jp
[2]Osaka Prefecture University, Osaka, Japan
tomoharu.nakashima@kis.osakafu-u.ac.jp
takuya.fukushima@edu.osakafu-u.ac.jp

Abstract. HELIOS2017 is a soccer simulation 2D team which has been participating in the RoboCup competitions since 2000. We recently focus on the improvement of action planning for players' decision making. In this paper, we proposed a pruning method in action planning using a clustering method and SVM in order to reflect the the team developper's intention in players' decision making.

Keywords: RoboCup · Soccer Simulation · Action Planning

1 Introduction

HELIOS2017 is a simulated soccer team for the RoboCup soccer 2D simulation league. The team has been participating in the RoboCup competition since 2000, and has won two championships [1]. We have released several open source software packages that help us to develop a simulated soccer team [2]

In previous years, we proposed a team formation model that uses Delaunay triangulation [**?**] and a multiagent planning framework [4]. They have been already available in the released software.

We recently focus on the improvement of action planning within tree search for the best next action. In this paper, we propose a pruning method to control the players' action planning without changing the evaluation function during the tree search process.

2 Pruning in Action Planning

In our previous works, we adopted a tree search method for decision making of our simulated players in order to plan tactical action sequences. In this section, we explain our pruning method for our search method.

2.1 Control of Action Planning by Pruning

In our plannning method, we use a cooperative action planning framework using a tree search method. This framework enalbles players to search sequential ball kicking actions among multiple teammate players. We define these sequential actions as a cooperative action plan.

The framework uses the best first search algorithm. A lot of action plans are generated during the search process and the best action plan is selected based on the evaluation value. The decision of players is highly depends on the evaluation function, that computes the evaluation value of each action plan. We have to design an appropriate evaluation function in order to select the action plan corresponds to the team strategy and tactics. However, it is difficult to find an appropriate evaluation function because we have to consider many possible feature values through trial-and-error.

Instead of findng the best evaluation function, we use a pruning approach to restrict actions generated in tree search process. This approach enables to control players' action pattern without adjusting the evaluation function. We propose a pruning method using support vector machine classification.

Discretization of Positional Information In our approach, a tactic is represented by an action sequence. In order to clarify position transition of actions, two continuous variables, x and y that represent the coordinate values in the soccer field, are discretized. We divide the soccer field into an $n \times m$ grid. In the grid field, if the ball exists in a grid cell, the grid cell takes a value 1, and the other grid cells take a value 0. In this way, the ball positions are represented by an $n \times m$ dimentional vector.

Pruning using Support Vector Machine(SVM) In our approach, action sequences not intended by the team developer are pruned during tree search process. The pruning is determined by a classifiler of support vector machine. The labeling method to create training data set is described in 2.2. An SVM classifier is generated by the created training data set. After pruning process, only action sequences representing the intended tactics by the team developer remain. The input to the SVM classifier is a discretized coodinate value of position where kicking action is performed.

2.2 Labeling action sequences using GUI

Since SVM is a supervised learning method, we need a training data set to acquire a classifier model. We extract action sequeces from game log files. Then, the team developers set a label to the extracted action sequences using a GUI if that is suitable for their intention. This labeled action sequence is used as a training data. We propose to apply a clustering method in order to classify similar action sequences. This approach reduces the human's action selection procedure. We use Gaussian mixture and EM algorithm with BIC as a clustering method. Figure 1 shows our GUI application. Action sequences orgnaized into one cluster are displayed in the main window. Action sequences intended by the team developer are labeled "1", other action sequences are labeled "-1".

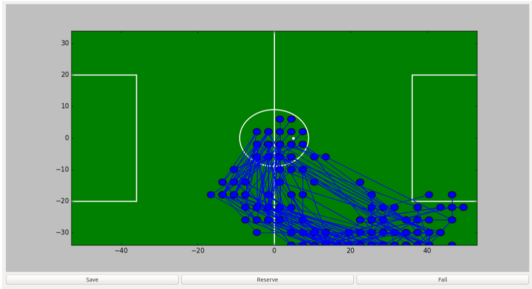

Fig. 1. Screenshot of the developed GUI

2.3 Experiments

At first, we perform an experiment to evaluate classification performance of our SVM classifier. Then, in order to evaluate the effectiveness of our pruning approach, we compare the performance of two teams, HELIOS2016 with the proposed model and the original HELIOS2016.

In this experment, the feature vector to be input to the SVM classifier is the discretized kick position, described in 2.1.

Experiment 1: SVM classifier In order to evaluate our SVM classifier, we performed simulation games against 8 teams, Shiraz, FURY, CSU_Yunlu, Info-Graphics, Ziziphus, HERMES, Ri-one and agent2d. 100 games were performed for each team. Then, extracted action sequences are classified by clustering algorithm. Classified action sequences are labeled by the team developper using our GUI application described in 2.2. Linear kernel and RBF kernel are used as the kernels of SVM. We applied 10-fold cross validation to the obtained training data set.

We compared the two types of grid resolutions, 7×7 and 23×26. Table 1 shows the result of 7×7, and Table 2 shows the result of 23×26. These results shows the linear kernel is better than the RBF kernel and the 23×26 grids is better than the 7×7 grids.

Table 1. Accurasy rate of SVM (7×7)

Kernel	Depth : 1	Depth : 2	Depth : 3	Depth : 4
Linear	80.31	81.98	83.65	84.60
RBF	67.93	67.93	67.93	67.93

Experiment 2: Control of Action Planning by Pruning In order to evaluate our pruning approach, we performed simulation games against 8 teams. We

Table 2. Accurasy rate of SVM (23×26)

Kernel	Depth : 1	Depth : 2	Depth : 3	Depth : 4
Linear	83.44	85.08	86.58	87.45
RBF	67.93	67.93	67.93	67.93

used the classifire model with linear kernel and 23×26 grid field. Figure 2 through 9 show the resulting pass courses of two teams, HELIOS2016 with the proposed method and original HELIOS2016. The left image of each figure shows the result of original team and the right side shows the result of proposed method. We can find that HELIOS2016 with the proposed method has a stronger tendency to pass on one side compared with the original team.

3 Conclusion

This paper described the research focus and the current effort of HELIOS2017. We proposed a pruning method in action planning using a clustering method and SVM in order to reflect the the team developper's intention in players' decision making. In the experiments, we evaluated the SVM classifier and the proposed pruning method. The results showed the proposed method enables us to control players' decision making without changing the evaluation function.

References

1. Hidehisa Akiyama and Tomoharu Nakashima. HELIOS2012: RoboCup 2012 Soccer Simulation 2D League Champion *RoboCup-2012: Robot Soccer World Cup XVI*, Springer Verlag, 2013
2. Hidehisa Akiyama and Tomoharu Nakashima. Helios base: An open source package for the robocup soccer 2D simulation. *Proc. of RoboCup Symposium 2013*, 2013
3. Hidehisa Akiyama and Itsuki Noda. Multi-Agent Positioning Mechanism in the Dynamic Environment. *RoboCup 2007: Robot Soccer World Cup XI*, 2008.
4. Hidehisa Akiyama and Tomoharu Nakashima and Shigeto Aramaki. Online Cooperative Behavior Planning using a Tree Search Method in the RoboCup Soccer Simulation *Proceedings of 4th IEEE International Conference on Intelligent Networking and Collaborative Systems (INCoS-2012)*, 2012.

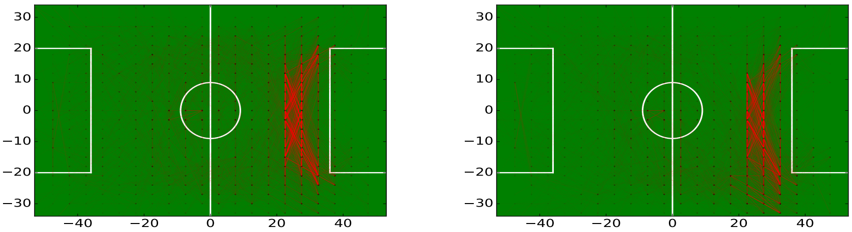

Fig. 2. Pass distribution against Shiraz

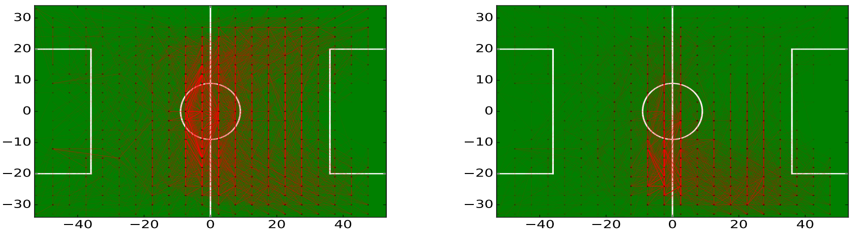

Fig. 3. Pass distribution against FURY

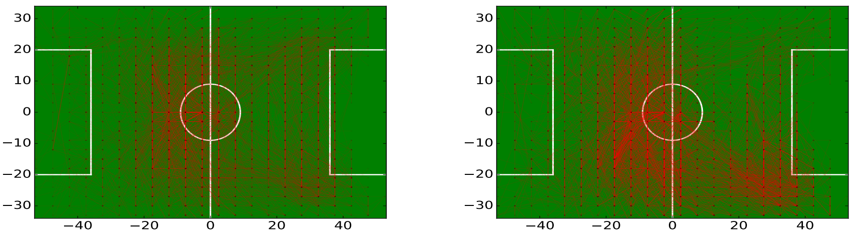

Fig. 4. Pass distribution against CSU_Yunlu

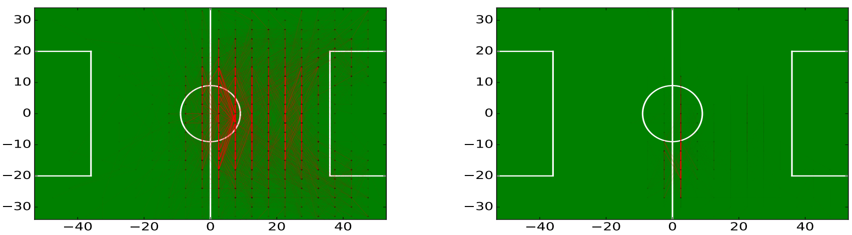

Fig. 5. Pass distribution against InfoGraphics

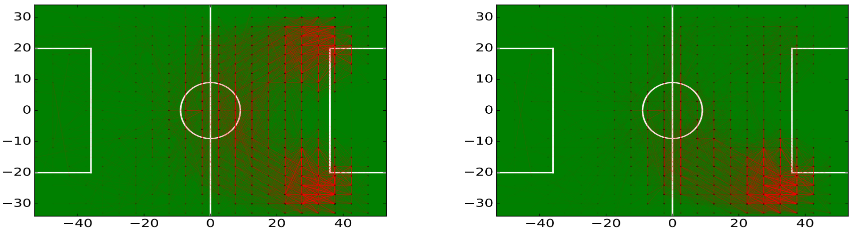

Fig. 6. Pass distribution against Ziziphus

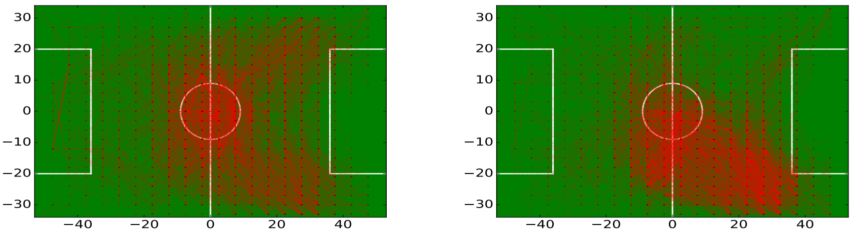

Fig. 7. Pass distribution against HERMES

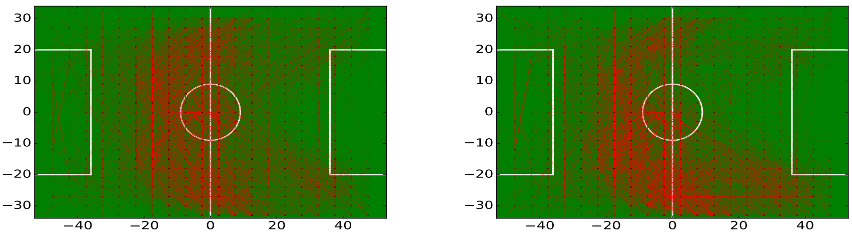

Fig. 8. Pass distribution against Ri-one

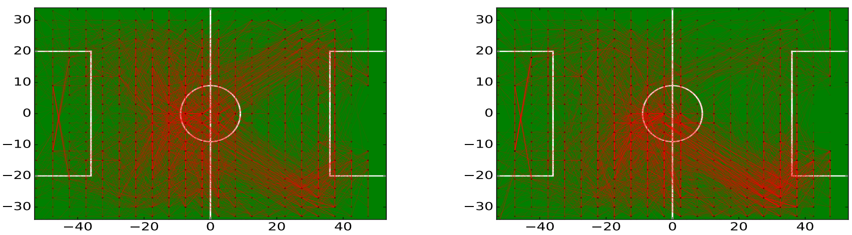

Fig. 9. Pass distribution against agent2d