Improvement of the AI system by analyzing characteristics in soccer simulator

Ryohei Watanabe s1220246 Supervised by Prof. Maxim Mozgovoy

Abstract

This paper deals with analyzing features by investigating various attributes for improvement of the AI system in soccer simulator. This study investigate effect of performance by attributes change used classifiers. We adopt the ball possession of agents (soccer players) as an index of performance in the soccer AI system. This experiments automatically measured whether increase or not the ball possession by repeatedly changing various settings. As a result, we show that it affect performance when selecting each configurations. Therefore, in our study, it turned out that the soccer AI system improved under what attributes.

1. Introduction

The technology of artificial intelligence is developing every year. It is the same in the field of sports games. The final goal of our project is to create an AI that behaves like a human being. Our project is making AI using actual professional league soccer match data and game data, but in order to achieve even better results, improvement of the existing system itself is necessary. Agents is soccer player operated by the system in soccer game simulation. It has parameters to make its characteristics controllable. The parameters of various attributes affecting the agent are set in the soccer game simulator we are using. In future research we will be able to produce better experimental results by using optimized these attributes. Thus, the purpose of this study is to optimize the parameters of the attributes to achieve the best performance when simulating based on what humans played the game. The criteria of hi performance is of the ball possession agents in this study.

2. Method

2.1 Game and AI

Our approach uses a simplified five-a-side 2D soccer simulation game (see Figure 1). This soccer simulation game is possible to learn the data we play once and simulate them. The soccer AI system follows the philosophy of TruSoft's Artificial Contender AI middleware[1], and makes use of TruSoft's AI SDK. Thus, the soccer agent acts according to the general scheme described in [2]. Its capabilities are sufficient to test the quality of the proposed AI solution[3;4]. This simulator game operates in two modes. In this study, we use PWB mode. This mode is Player With Ball, the mode where we always control the player who has the ball. Each agents moves only the person corresponding to the player holding the ball, and the other agents stop.

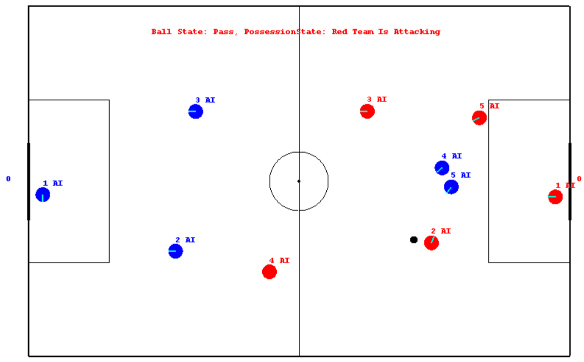

Figure 1. Five-a-side 2D soccer simulator

2.2 Zoom levels and classifiers

Zoom levels are various level of abstractions used in our system. The basic unit used to match a similar game situation is a "classifier" or an "attribute", such as ball X coordinate or ball state. The list of classifiers used to match the situation is one zoom level. When we remove some of the classifiers, we get a higher zoom level and so on. For Example,

Z0 (Zoom level 0) = Player1_X, Player1_Y, Ball_X, Ball_Y, BallState
Z1 = Ball_X, Ball_Y, BallState
Z2 = BallState

Zoom level N should have fewer factors than Zoom level N-1, Therefore if we switch off some factor on Z0, we should also switch off it on Z1-Z3. The zoom levels are designed in such a way that we will need to switch on or off the corresponding classifiers on all zoom levels at once. If we want to switch off the classifier "Player1_X0", just set Enable_PlayerX0 to false.

2.3 Experimental procedure

We try to change different configurations and investigate how they affect the game performance in the soccer simulator. In this experiment, we adopted the ball possession as an indicator of performance. In the general case, such an approach should be based on the assumption that the each attributes are dependent each other. Therefore, it is often tried to make all combinations of those attributes. However, if there are N settings, the method must perform a maximum of 2^N tests. It is not realistic when the number of parameters of setting is large. Moreover, because the attributes handcrafted in this case, this approach can be based on the assumption that the attributes are independent. Therefore, we conducted the experiment by two methods as follows:

- **Preprocessing:** Play a game for five minutes and save this recording. After that, we set the time we want to learn to exclude extra parts, such as when no one is touching the ball at the beginning of the game. Also set the simulation time to 7500 frame (about 5 minutes).
- **Step 1:** Generate configuration file and change parameters of configuration one by one. First time, we set to all true. Second time, we change only Enable_PlayerX3 to false. (Because if you disable Enable_PlayerX1, we have to disable also Enable_PlayerX2 and Enable_PlayerX3.) From next time, if we change something false and the behavior becomes worse, there is no reason to continue this line. Because we already made the system worse. If we change something false and the behavior becomes better, then we'll change next parameter from true to false to investigate this branch.
- **Step 2:** make agents learn behaviors of players in the game saved by learning tool and reflect changing configuration to the simulator.
- **Step 3:** Simulate game for 7500 frames (about five minutes).
- **Step 4:** Calculate all possession of the home team(percentage of home team owning a ball). In the first method (referred to below as Method 1), proceed to step 5. In another method (referred to below as Method 2), if we simulate from one to nine times with the same settings, return to step 3. If we simulate ten times with the same settings, return to step 5.
- **Step 5:** Check statistics and judge performance whether better or worse. After this operation, return to step 1.

In Method 1, we compare each simulation of 7500 frames. In Method 2, every time ten simulations are performed with the same settings, calculate those average ball possession and compare to the previous average ball possession. We created a software to run repeatedly Method 1 and Method 2. When this both programs reach last line of Enable_Player5X, the one test finish.

3. Result

Then let's us first see the results in method 1. In method 1, 20 tests were done. Table 2 and Table 3 show the results of each tests from Test 1 to Test 10 tests and from Test 11 to Test 20 in Method 1. “All true” is a state where there is no false and everything is true. Highest best ball possession at the last point in time is 76% in Test 6. Lowest best ball possession at the last point in time is 65% in Test 7. It is 11% wide from the lowest to the highest. Average best ball possession at the last point in time is 70.65% in 20 tests.

Next, we see the results in method 2. Table 1 shows results of tests in Method 2. To explain items of the table, “Classifiers” is classifiers when changed from true to false. “Avg possess” is average ball possession when running different runs ten times with the same configuration. “Best possess” is best ball possession at each measurement point. Best ball possession at the last point in time is 64.6%. “Possess range” is the range of the percentages from lowest ball possession to highest ball possession when running different simulations ten times with the same configuration. Average of Possession range is 13.65%.

Table 1
Details of experimental results in Method 2

Classifiers	Avg possess	Best possess	Possess range
All true	60.4%	60.4%	15%
PlayerX3	60.2%	60.4%	14%
PlayerY3	61.6%	61.6%	11%
PlayerY2	63%	63%	10%

PlayerY1	63.4%	63.4%	10%
PlayerY0	59.5%	63.4%	11%
PlayerNum-ber3	61.8%	63.4%	15%
BallState2	62.9%	63.4%	11%
Danger-MoveFor-ward0	58.2%	63.4%	13%
Movement-Direction2	62.9%	63.4%	11%
ClosestThrea tDistance2	64.6%	64.6%	14%
ClosestThrea tDistance1	61.6%	64.6%	5%
ClosestThrea tDirection2	58.2%	64.6%	17%
IsPWB3	62.3%	64.6%	19%
PdeFromP-wb2	60.3%	64.6%	18%
Player4X2	63.6%	64.6%	22%
Player5X2	61%	64.6%	16%

4. Discussion

Let's us first discuss the results in method 1. Figure 2 shows the number of times ball performance was increased when changing classifier from true to false, in other words the number of times the performance improved at that time during 20 times in Method 1. As a result, when changing classifier from true to false, the number of times the performance got better was half that was not more than 10 times. However, it is premature to judge that all attributes should be set to true by looking at only that fact. Although it did not exceed half the number of times, it is obviously possible to see an opening in the number of times difference in each attribute. It is estimated that this was due to some impact on its performance. "Enable_Player1X3" and "Enable_BallState2" clearly differ from other ones. It is presumed that there is some influence that improves performance by changing these to false. Even when I set the same setting, the ball possession differ. Best ball possession at the last point in time is from 65% to 76% and average is 70.65%. This is a result of good accuracy in Method 1. It is said that the performance has improved since the ball possession of this degree was obtained.

Next, let's discuss the results in Method 2. Looking at “range of possess” in Table 1, it can be seen that there is difference of from 5% to 22% across different runs even with the same configuration. This results mean in order to collect accurate values, it is necessary to repeat the tests many times with the same configuration. Therefore, It is difficult to repeat all combinations of attributes in maximum many times in general method. However, when the difference is small in like case of “Enable_ClosestThreatDistance1”, There is a possibility that an accurate value can be obtained even with less frequent trials. Even in this Method 2, “Avg possess” of “Enable_PlayerX3” and “Enable_BallState2” is relatively high, so it is highly likely that we should set those parameters to false. By repeating method 2, we will be able to gather more accurate statistics.

5. Conclusion

In this paper, we show that how it affects the game performance by changing each attributes and analyzing. While the results are still preliminary, they clearly show that we will be able to take more accurate statistics by executing repeatedly this software we created. We hope that this work will help to build high quality AI systems for soccer games that are able to mimic individual human behavior.

Reference

[1] TruSoft. Artificial Contender: bringing more life into gameplay with behavior-capture AI. http://www.trusoft.com/

[2] M. Mozgovoy, I. Umarov. Behavior Capture: Building Believable and Effective AI Agents for Video Games. International Journal of Arts and Sciences, 2011, vol. 4(20).

[3] M. Mozgovoy and I. Umarov, "Building a Believable Agent for a 3D Boxing Simulation Game," Proc. of the 2nd International Conference on Computer Research and Development, 2010, pp. 46-50.

[4] M. Mozgovoy, I. Umarov. Believable Team Behavior: Towards Behavior Capture AI for the Game of Soccer. Proceedings of the 8th International Conference on Complex Systems, Boston, USA, 2011, pp. 1554-1564.

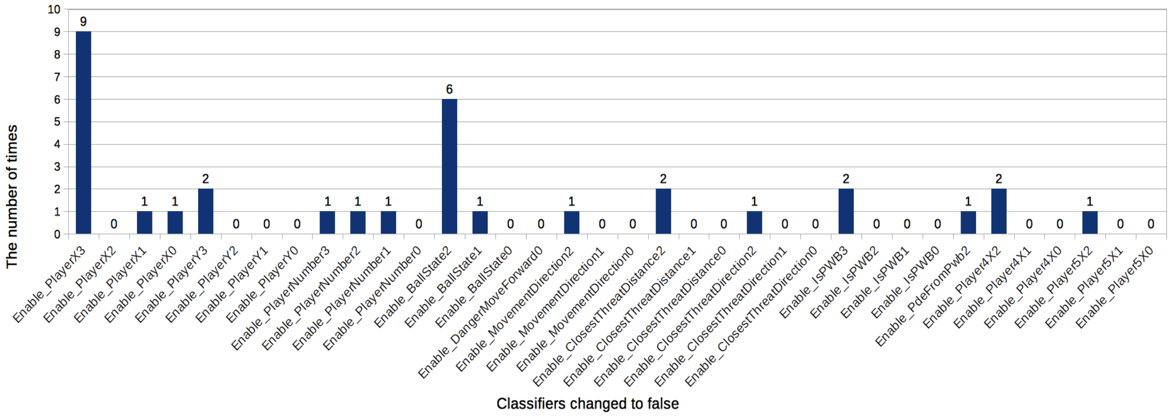

Figure 2. The number of times during 20 times judged performance better when the classifiers was changed from true to false in Method 1

Table 2. Detail of Experimental results in Method 1

Classifiers changed to false	Test 1		Test 2		Test 3		Test 4		Test 5		Test 6		Test 7		Test 8		Test 9		Test 10	
	Ball possession when changed to false	Best possession at each measurement point																		
All true	59%	59%	64%	64%	57%	57%	64%	64%	69%	69%	63%	63%	65%	65%	67%	67%	67%	67%	57%	57%
PlayerX3	66%	66%	63%		68%	68%	59%		63%		72%	72%	64%		63%		63%		69%	69%
PlayerX2	63%				65%						56%								61%	
PlayerX1																				
PlayerX0																				
PlayerY3	61%		63%		67%		62%		55%		76%	76%	59%		67%		61%		55%	
PlayerY2											66%				65%					
PlayerY1																				
PlayerY0																				
PlayerNumber3	58%		60%		66%		57%		66%		62%		64%		59%		60%		68%	
PlayerNumber2																				
PlayerNumber1																				
PlayerNumber0																				
BallState2	56%		74%	74%	61%		58%		62%		58%		56%		62%		36%		60%	
BallState1			61%																	
BallState0																				
DangerMoveForward0	58%		56%		58%		59%		64%		67%		62%		57%		62%		56%	
MovementDirection2	65%		63%		65%		67%	67%	62%		54%		61%		62%		58%		58%	
MovementDirection1							54%													
MovementDirection0																				
ClosestThreatDistance2	68%	68%	68%		58%		70%	70%	61%		65%		62%		59%		63%		49%	
ClosestThreatDistance1	54%						59%													
ClosestThreatDistance0																				
ClosestThreatDirection2	61%		62%		64%		67%		66%		66%		60%		67%	67%	61%		57%	
ClosestThreatDirection1															56%					
ClosestThreatDirection0																				
IsPWB3	68%	68%	68%		58%		62%		60%		62%		63%		64%		57%		53%	
IsPWB2	53%																			
IsPWB1																				
IsPWB0																				
PdeFromPwb2	66%		63%		61%		65%		67%		66%		57%		62%		60%		65%	
Player4X2	64%		63%		69%	69%	56%		58%		64%		62%		64%		58%		59%	
Player4X1					60%															
Player4X0																				
Player5X2	57%		65%		57%		61%		63%		66%		58%		60%		61%		55%	
Player5X1																				
Player5X0																				
Final Best possession		68%		74%		69%		70%		69%		76%		65%		67%		67%		69%

Table 3. Detail of experimental results in Method 1

Test 11		Test 12		Test 13		Test 14		Test 15		Test 16		Test 17		Test 18		Test 19		Test 20		Count better changed to false	Classifiers changed to false
Ball possession when changed to false	Best possession at each measurement point																				
63%	63%	58%	58%	66%	66%	69%	69%	61%	61%	62%	62%	69%	69%	63%	63%	65%	65%	54%	54%		All true
71%	71%	53%		61%		66%		73%	73%	63%	63%	63%		66%	66%	62%		56%	56%	9	PlayerX3
64%								58%		62%				59%				56%		0	PlayerX2
																		62%	62%	1	PlayerX1
																		64%	64%	1	PlayerX0
59%		60%	60%	61%		63%		68%		58%		59%		60%		64%		58%		2	PlayerY3
		54%																		0	PlayerY2
																				0	PlayerY1
																				0	PlayerY0
64%		61%	61%	53%		61%		68%		55%		64%		63%		64%		62%		1	PlayerNumber3
		62%	62%																	1	PlayerNumber2
		66%	66%																	1	PlayerNumber1
		62%																		0	PlayerNumber0
64%		66%	66%	68%	68%	55%		66%		72%	72%	74%	74%	61%		71%	71%	55%		6	BallState2
		69%	69%	64%						62%		67%				65%				1	BallState1
		54%																		0	BallState0
62%		62%		60%		61%		63%		52%		61%		61%		58%		56%		0	DangerMoveForward0
67%		64%		63%		60%		62%		57%		66%		64%		62%		63%		1	MovementDirection2
																				0	MovementDirection1
																				0	MovementDirection0
61%		53%		62%		66%		63%		70%		62%		56%		53%		57%		2	ClosestThreatDistance2
																				0	ClosestThreatDistance1
																				0	ClosestThreatDistance0
64%		54%		66%		63%		64%		65%		55%		61%		67%		60%		1	ClosestThreatDirection2
																				0	ClosestThreatDirection1
																				0	ClosestThreatDirection0
57%		76%	76%	67%		60%		69%		67%		56%		60%		54%		62%		2	IsPWB3
		57%																		0	IsPWB2
																				0	IsPWB1
																				0	IsPWB0
66%		58%		56%		72%	72%	62%		63%		65%		59%		64%		60%		1	PdeFromPwb2
61%		67%		64%		66%		70%		61%		8%		71%	71%	66%		64%		2	Player4X2
														61%				62%		0	Player4X1
																				0	Player4X0
66%		63%		64%		57%		67%		67%		64%		66%		70%		71%	71%	1	Player5X2
																		59%		0	Player5X1
																				0	Player5X0
	71%		76%		68%		72%		73%		72%		74%		71%		71%		71%		Final Best Possession