Analyzing and Learning an Opponent's Strategies in the RoboCup Small Size League

Kotaro Yasui, Kunikazu Kobayashi, Kazuhito Murakami, and Tadashi Naruse

Graduate School of Information Sciences and Technology,
Aichi Prefectural University, Japan
im121007@cis.aichi-pu.ac.jp,
{koboyashi,murakami,naruse}@ist.aichi-pu.ac.jp

Abstract. This paper proposes a dissimilarity function that is useful for analyzing and learning the opponent's strategies implemented in a RoboCup Soccer game. The dissimilarity function presented here identifies the differences between two instances of the opponent's deployment choices. An extension of this function was developed to further identify the differences between deployment choices over two separate time intervals. The dissimilarity function, which generates a dissimilarity matrix, is then exploited to analyze and classify the opponent's strategies using cluster analysis. The classification step was implemented by analyzing the opponent's strategies used in set plays captured in the logged data obtained from the Small Size League's games played during RoboCup 2012. The experimental results showed that the attacking strategies used in set plays may be effectively classified. A method for learning an opponent's attacking strategies and deploying teammates in advantageous positions on the fly in actual games is discussed.

1 Introduction

Robotic soccer in the RoboCup Small Size League (SSL) involves a competition between teams of at most six robots on a 6050 mm by 4050 mm field. Two cameras are positioned 4 m above the field to collect photographs of the field every 1/60th of a second. The photographs are sent to a vision computer dedicated to image processing. The vision computer calculates the positions of the robots and the ball on the field and sends these coordinates to each team's computer. Each team then uses this information to compute the next positions each robot will move to based on their strategy. These positions are then sent to the robots via radio communication. A referee box computer, which controls the progression of the game, sends messages such as 'start throw-in', 'start corner kick', etc. to each team's computer. The game advances automatically without the intervention of any person except for the referees and the person controlling the referee box.

The moving speeds of robots in the SSL have increased year-on-year. For instance, the champion team in the RoboCup 2012 moved its robots at a maximum speed of 3.5 m/s [1], and the champion team passed the ball at speeds exceeding 4 m/s. In this environment, predicting the behavior of an opponent is very important.

S. Behnke et al. (Eds.): RoboCup 2013, LNAI 8371, pp. 159–170, 2014.

In recent SSL games, each team has tended to decide on a strategy depending on the positions and velocities of the teammates, the opponents, and the ball. The referee box's signals are also used to select a strategy. By contrast, in human soccer, each player decides his/her behavior using, in addition to the above four factors, the history of play such as "how the opponents have played during the current game." It is reasonable to expect that a robotic soccer game may also play out by selecting an appropriate strategy based on an analysis of the opponent's strategies or behaviors during the previous game plays.

In an effort to analyze and learn from an opponent's strategies, a dissimilarity function that identified differences between an opponent's deployment choices at two different instances was developed. The dissimilarity function was then extended to identifying the differences between deployment choices made during two different time intervals. The opponent's strategies were then analyzed by using this dissimilarity function. The dissimilarity matrix generated from the dissimilarity function was used to perform a cluster analysis to classify the opponent's strategies. This method was applied to the data logged during SSL games played during the RoboCup 2012. This method was shown to be able to effectively classify the attacking strategies during set plays. A method for learning an opponent's attacking strategies and deploying teammates in advantageous positions on the fly in actual games is discussed.

2 Related Work

Bowling et al. [2] proposed a method for implementing an opponent-adaptive play selection. This method, which has been used in the SSL games, is formulated in the frame of an experts problem or a k-armed bandit problem. The method selects an effective play from a playbook based on the "regret" measure[1]. The selected play is then given an appropriate reward corresponding to the result of the play: success, failure, completion, or abort. The regret measure is updated based on the accumulated reward of each play.

Trevizan and Veloso [3] have proposed a method for comparing the strategies of teams played in the SSL. In their method, a team's strategy is represented as an episode matrix, the elements of which are the means and the standard deviations of variables over the time segment S in the time series T, where T represents a game. They selected 23 variables for analysis, including the distance between each robot and the ball. A similarity function $s(\cdot, \cdot)$ was then defined using the matrices to represent the closeness between two episodes. The method was applied during defense episodes, and they discussed the closeness of the defense strategies used by the two teams.

Visser and Weland [4] proposed a method for classifying an opponent's behavior based on a decision tree using the data logged by the RoboCup Simulation League. They classified, for instance, the behaviors of a goalkeeper into three

[1] The regret is the amount of additional reward that could have been received if an expert selection algorithm had known which expert would be the best and had chosen that expert at each decision point[2].

categories: the goalkeeper backs up, the goalkeeper stays in the goal, and the goalkeeper leaves the goal. They also analyzed the pass behaviors of the opponents in a similar way.

3 Comparison of Strategies

In the proposed dissimilarity method, an opponent's strategies were first classified using a dissimilarity function that quantifies the difference between two deployment decisions made by an opponent at times t_1 and t_2. The dissimilarity function was defined on a field coordinate system having an origin at the center of the field, with the x-axis pointing toward the center of the opponent's goal mouth and the y-axis representing the center line.

Let $R_i(t_k)$ be the coordinates of the opponent i at time t_k $(k = 1, 2)$. Assume that m opponents are present on the field at time t_1 and n opponents at time t_2. (Robots are numbered from 1 to m at time t_1 and from 1 to n at time t_2.) The dissimilarity function d is then defined as follows:

$$d(t_1, t_2) = \min_{U \in \{U_1, U_2\}} \left\{ \min_{\sigma \in S_6} \sqrt{\mathrm{trace}(F(U)P_\sigma)} \right\}, \tag{1}$$

$$U_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \; U_2 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},$$

$$F(U) = [f_{ij}],$$

$$f_{ij} = \begin{cases} \|UR_i(t_1) - R_j(t_2)\|^2 & (1 \le i \le m, \; 1 \le j \le n) \\ \Delta^2 & (\text{otherwise}) \end{cases},$$

where S_6 is the symmetric group of degree six, P_σ is the permutation matrix of a permutation σ, and U_2 is the matrix describing a reflection transformation through the x-axis used to change the sign of the y-coordinate. F is a 6×6 matrix, the element f_{ij} of which is the Euclidean distance between the position of the opponent i at time t_1 and the position of the opponent j at time t_2, if robots i and j are on the field. If one (or neither) robot is not on the field, this element is assigned the value Δ^2. Δ indicates a virtual distance and is assigned a constant value here. The selection of Δ requires a careful process that will be discussed elsewhere. d indicates the sum of the distances between all corresponding robots when the minimal distance mapping between robots i at t_1 and j at t_2, for all i and j, is achieved.

The opponent's strategy is assumed to rely on the type of skill attributed to each robot. A strategy is implemented based on the skill assigned to each robot, which can vary between t_1 and t_2. The dissimilarity function d is expected to function as designed, even in the event that the robots are permuted. In the context of robotic soccer, all strategies may be assumed to be symmetric along the x-axis[2], and the reflection U_2 given in Equation (1) can be considered. The ball and teammates' positions are excluded from Eq. (1) for the following reasons:

- The robots' positions depend on the ball so that relative positions between robots are sufficient to characterize the behavior on the field;
- The positions of the teammates can negatively affect the learning process because the behavior of the teammates are affected by the learning process.

Next, Eq. (1) was used to define a dissimilarity function $d_1(\cdot,\cdot,\cdot)$ between the deployment choice of an opponent at time t_1 and a set of deployment choices made during a time interval $[T_s, T_e]$, as follows:

$$d_1(t_1, T_s, T_e) = \min_{t \in [T_s, T_e]} d(t_1, t). \tag{2}$$

Equation (2) may be used to define a dissimilarity function $d_2(\cdot,\cdot,\cdot,\cdot)$ between two sets of deployments: one for the time interval $[T_s^{(i)}, T_e^{(i)}]$ and the other for $[T_s^{(j)}, T_e^{(j)}]$, as follows:

$$d_2(T_s^{(i)}, T_e^{(i)}, T_s^{(j)}, T_e^{(j)}) = \min_{t \in [T_s^{(i)}, T_e^{(i)}]} d_1(t, T_s^{(j)}, T_e^{(j)}). \tag{3}$$

Equation (3) shows the dissimilarity function between the two most similar deployment choices in the two distinct sets. The opponent's strategies may then be classified by applying Eq. (3) to the sets of deployments. The next section discusses the use of cluster analysis in classifying the opponent's strategies based on the dissimilarity function given by Eq. (3).

4 Cluster Analysis

Consider the N sets of deployment choices, each of which comprises a set of deployments made within a time interval. Equation (3) is then applied to any two sets in the N sets of deployments to calculate a dissimilarity matrix of size $N \times N$, the elements of which are the dissimilarities between the set plays. The matrix may be regarded as a proximity matrix, and a cluster analysis may be used to classify the opponent's strategies.

4.1 Clustering Methods

Many clustering methods are available for use in cluster analysis[8]. Typical methods include:

k-means Algorithm. The k-means algorithm is a cluster analysis method that aims to partition n observations into k clusters such that each observation belongs to the cluster characterized by the nearest mean. This creates a partitioning of the data space into Voronoi cells.

[2] In human soccer, a player may have his/her strong wing; however, in robotic soccer (especially in the SSL), none of the robots has a strong wing, in general. Therefore, it is generally acceptable to assume symmetric positions.

Ward's Method. Ward's method is a hierarchical cluster analysis method that shares many of the features of variance analysis. A linkage (of clusters) function, which specifies the distance between two clusters, is computed as an increase in the "error sum of squares" (ESS) after two clusters are fused into a single cluster. Ward's Method seeks to select successive clustering steps that minimize the increase in ESS at each step.

Group Average Clustering. The group average clustering method is a hierarchical method in which the distance between two clusters is calculated based on the average distance between all pairs of objects in the two different clusters.

Both the k-means algorithm and Ward's method are practical methods; however, they cannot be used to calculate the centroid of a cluster. In this problem, it is difficult to calculate the centroid because each element in a cluster denotes a continuous deployment choice. The group average clustering method, on the other hand, calculates the distance between objects (deployments in this case), and this distance is sufficient for implementing the clustering method. The group average clustering method was therefore chosen.

4.2 Estimating the Number of Clusters

The group average clustering method creates a hierarchical structure of clusters; however, the method does not estimate the number of clusters. To do so, the Davies–Bouldin index (DBI)[5] was used. The DBI for K clusters is defined as follows:

$$\mathrm{DB}(K) = \frac{1}{K}\sum_{i=1}^{K} \max_{j \neq i} \frac{S_i + S_j}{M_{ij}}, \tag{4}$$

where M_{ij} is a measure of the **separation** between two clusters C_i and C_j, and S_i is a measure of **cohesion** within a cluster C_i. M_{ij} and S_i may be defined freely under some constraints[5]. The optimal number of clusters is given by K, which is selected to minimize $\mathrm{DB}(K)$ over the range of K identified by some criterion, for example, using Sturges' formula . The definition of the DBI requires the judicious selection of an appropriate range for K because $\mathrm{DB}(K)$ approaches 0 as the number of single object clusters increases.

Equation (3) is then used to define the separation M_{ij} and cohesion S_i as follows:

$$S_i = \frac{1}{|C_i|(|C_i|-1)} \sum_{X_k \in C_i} \left\{ \sum_{X_l \in C_i, X_l \neq X_k} d_2(T_s^{(k)}, T_e^{(k)}, T_s^{(l)}, T_e^{(l)}) \right\},$$

$$M_{ij} = \frac{1}{|C_i||C_j|} \sum_{X_k \in C_i} \sum_{X_l \in C_j} d_2(T_s^{(k)}, T_e^{(k)}, T_s^{(l)}, T_e^{(l)}),$$

where S_i is the mean distance between any two objects in C_i and M_{ij} is the distance between C_i and C_j, computed using the group average clustering method. The values of S_i and M_{ij} satisfy the constraints given in [5].

5 Experimental Results

Many goals in the SSL games are scored based on the set plays, such as the identification of a throw-in, corner kick, or goal kick. Each team implements a variety of strategies in the set plays. The method proposed in the previous sections was applied here to the set plays executed by the teams participating in the RoboCup 2012 games in an effort to analyze and classify the opponents' strategies.

Let X_i be a set of deployment choices made during the i-th set play ($1 \leq i \leq N$). The time interval surrounding X_i is $[T_r^{(i)}, T_e^{(i)}]$, where $T_r^{(i)}$ is the time the team receives a start command for the set play from the referee box and $T_e^{(i)}$ is the time at which a kick is carried out in the set play.

$T_s^{(i)}$ can be defined as

$$T_s^{(i)} = \max(T_e^{(i)} - T_{behavior}, T_r^{(i)}), \tag{5}$$

where $T_{behavior}$ is a constant that specifies the time at which the robot takes an action. Small $T_{behavior}$ values are sufficient for an expert team because their robots move fast. Somewhat larger $T_{behavior}$ values absorb the dispersion of the set plays based on a strategy. For unfamiliar teams, the assumption of a small $T_{behavior}$ value is recommended. In this paper, $T_{behavior}$ is set to 1.0 sec.

Equation (3) was used to compute the dissimilarity functions

$$d_2(T_s^{(i)}, T_e^{(i)}, T_s^{(j)}, T_e^{(j)}),\ (1 \leq i \leq N,\ 1 \leq j \leq N). \tag{6}$$

A dissimilarity matrix[3] was then generated, and a dendrogram was calculated based on the group average clustering. Finally, the number of clusters present was estimated using Equation (4). The opponents' strategies used in the set plays among the K strategies were then classified. The following Sturges' formula[6] was used to estimate the range of K:

$$1 \leq K \leq \lceil \log_2 N + 1 \rceil, \tag{7}$$

where $\lceil x \rceil$ is the ceiling function of x.

5.1 Classifying RoboDragons' Strategies

First, the RoboDragons' strategies (our team's strategies) were classified using the proposed method. RoboDragons used four attacking strategies during the set plays that took place in the RoboCup 2012 world championship. These strategies were denoted $A_i, (1 \leq i \leq 4)$. The RoboDragons' simulation system[4] was used

[3] In the experiments described in this section, Δ in Eq. (1) was not used because six robots on each team were always on the field.

[4] The proposed method was first applied to the real games played during the RoboCup 2012. These games were easy to classify because the parameter values used to implement the strategies were fixed. The simulation system was then used to execute set plays in which the values of the parameters were varied.

to execute a pseudo-game involving RoboDragons (Blue) vs. RoboDragons (Yellow)[5]. Twenty-four set plays in total were executed. One strategy was used in each set play.

Each strategy was used six times over the course of the 24 games. According to the rules of the SSL, the ball was placed at the (x, y) coordinates at the start of a set play, where the y-coordinate is 1915 mm and the x-coordinate is randomly selected from within a range of values that permits execution of the set play.

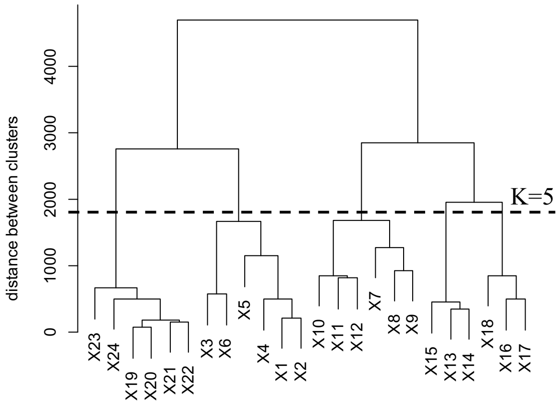


Fig. 1. Dendrogram (RoboDragons)

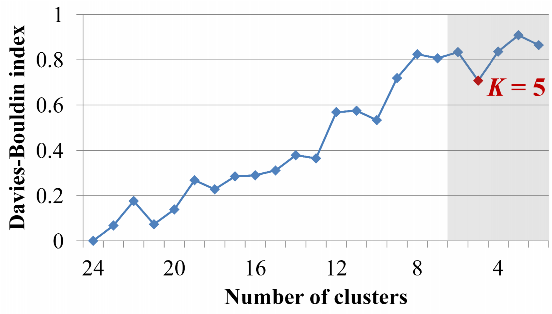


Fig. 2. Davies–Bouldin index (RoboDragons)

Figure 1 shows a dendrogram of the experimental results obtained[6], and Figure 2 shows the Davies–Bouldin index. The number of clusters fell within the range $1 \leq K \leq 6$, according to Equation (7). Figure 2 shows that an estimate of $K = 5$ in this case was reasonable. The dendrogram shown in Fig. 1 was cut off at the height at which the number of clusters reached five to obtain the following five clusters.

$$C_1 = \{X_1, X_2, X_3, X_4, X_5, X_6\}$$
$$C_2 = \{X_7, X_8, X_9, X_{10}, X_{11}, X_{12}\}$$
$$C_3 = \{X_{13}, X_{14}, X_{15}\}$$
$$C_4 = \{X_{16}, X_{17}, X_{18}\}$$
$$C_5 = \{X_{19}, X_{20}, X_{21}, X_{22}, X_{23}, X_{24}\}$$

The set plays implemented in C_1 used the A_1 strategy, C_2 used A_2, and C_5 used A_4. The set plays characterized by the strategy A_3 could be classified as belonging to one of two clusters, C_3 and C_4. If $K = 4$, however, then C_3 and C_4

[5] Blue and Yellow are the colors used to identify the teams in the SSL.

[6] The dendrograms in this paper were drawn using the R statistical software package (`http://www.r-project.org/`).

were unified, and the clusters correctly separated the strategies. These results revealed the utility of the proposed method. Figure 1 showed that the strategy A_4 was easier to identify than the other strategies because the height of C_5 was the smallest. (Strategy A_3 was characterized as the height of $(C_3 \cup C_4)$.)

5.2 Classifying the Opposing Teams' Strategies

This section discusses attempts to classify the opposing teams' strategies. The data logged during the final game of the RoboCup 2012, in which Skuba (Blue) opposed ZJUNlict (Yellow), were analyzed here. In the game, 62 set plays were restarted from 10 cm inside of the touch boundary crossed by the ball. Of these set plays, 37 were implemented by Skuba and 25 were implemented by ZJUNlict. Figures 3 and 4 show the dendrograms of the set plays implemented by Skuba and ZJUNlict, respectively. Equation (4) was used to estimate the number of clusters: $K = 5$ for Skuba, and $K = 6$ for ZJUNlict.

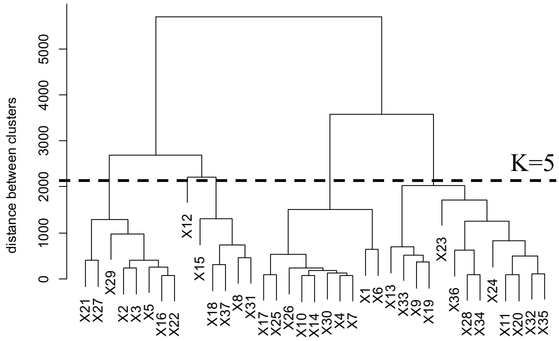


Fig. 3. Dendrogram (Skuba)

The analysis of Skuba's strategies will be discussed first. Figure 5 shows the classified results. Each image illustrates a deployment choice made immediately after the kick was taken during a set play. In these images, generated by our logged data review system, the sizes of the ball and robots are enlarged, and the number in each robot is the robot's ID. Note that the attacking direction changed from set play X_{17} because the teams changed sides after half time.

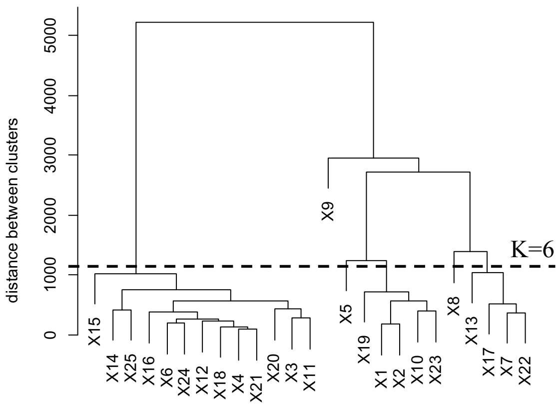


Fig. 4. Dendrogram (ZJUNlict)

An analysis of the clusters readily identified the strategies used. The strategies were found to be characterized as follows:

- C_1 encompassed a strategy in which the ball was kicked directly toward the goal without passing between robots.
- C_2 encompassed a strategy in which the ball was passed to a teammate at the far side of the opponent's goal area after a corner kick had been taken.

- C_3 encompassed a strategy in which the ball was passed to the teammate at the near side of the center line after a throw-in had been taken.
- C_4 encompassed a strategy in which the ball was passed to the teammate at the far side of the field. This strategy resembled the strategy C_1.
- C_5 encompassed a strategy in which the ball was passed to the teammate at the center of the field.

The analysis of ZJUNlict's strategies will be discussed next. The following classification results were obtained from this analysis. (The corresponding images are omitted due to limitations on the article space.)

$$
\begin{aligned}
C_1 &= \{X_1, X_2, X_{10}, X_{19}, X_{23}\} \\
C_2 &= \{X_3, X_4, X_6, X_{11}, X_{12}, X_{14}, X_{15}, X_{16}, X_{18}, X_{20}, X_{21}, X_{24}, X_{25}\} \\
C_3 &= \{X_5\} \\
C_4 &= \{X_7, X_{13}, X_{17}, X_{22}\} \\
C_5 &= \{X_8\} \\
C_6 &= \{X_9\}
\end{aligned}
$$

The strategies could be characterized as follows:

- C_1 encompassed a strategy in which the ball was passed to the teammate at the far side of the opponent's goal area.
- C_2 encompassed a strategy in which short passes were made to the teammate located along the direction of the goal. The team used this strategy many times.
- C_3 was similar to strategy C_1.
- C_4 encompassed a strategy in which the ball was passed to the teammate at the far side of the field after a throw-in had been taken on the opponent's side.
- C_5 was similar to strategy C_4.
- C_6 appeared to be similar to strategy C_4, although the placement of two robots on opposite sides of the ball may have led to the use of another strategy.

The results of this experiment revealed that the classification of an opponent's strategies was possible.

6 Application: On-line Learning

Section 5 demonstrated that an opponent's strategies could be classified using the method proposed here. This method was applied to an on-line learning algorithm to assist in selecting an advantageous action during the opponent's $(N + 1)$th

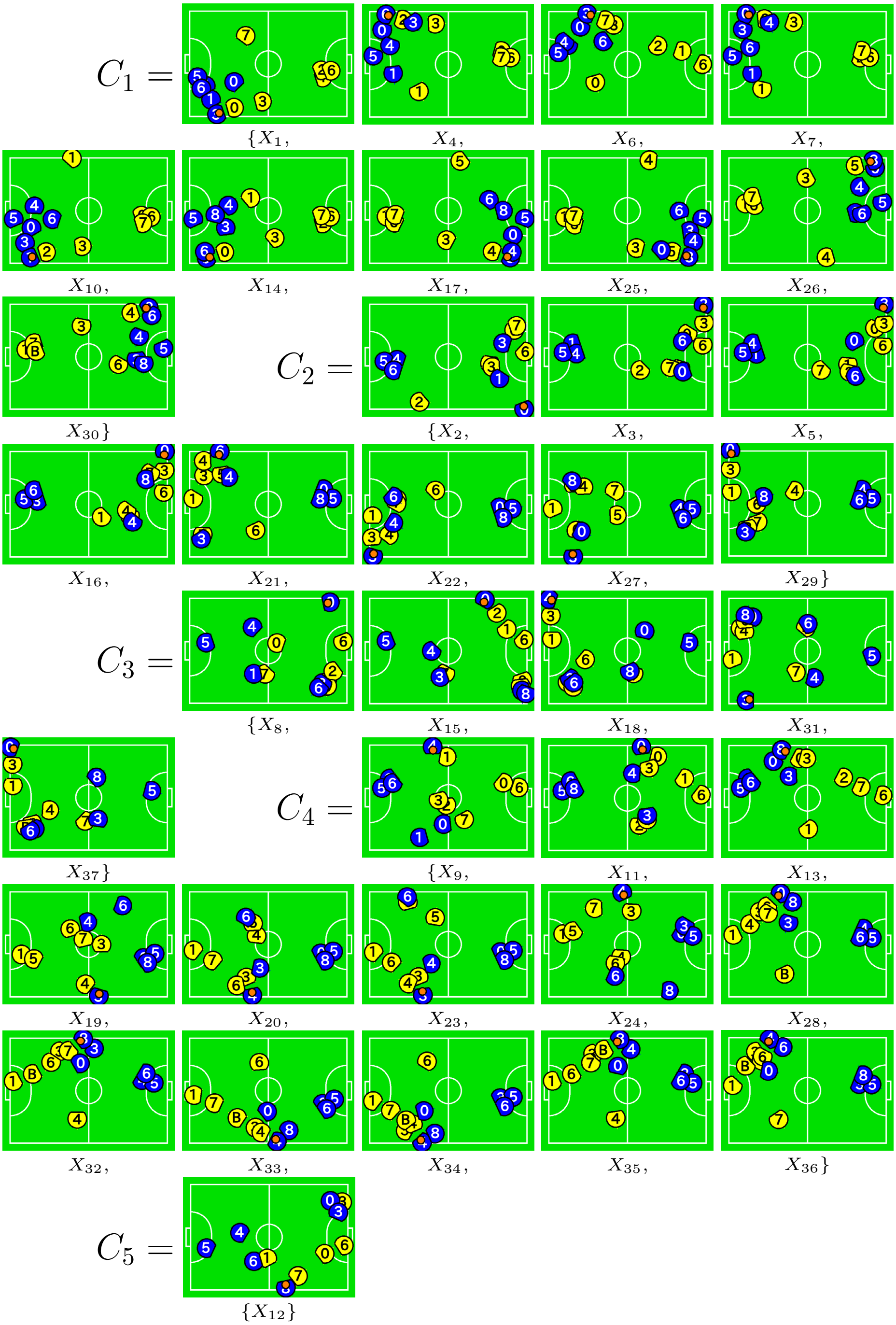
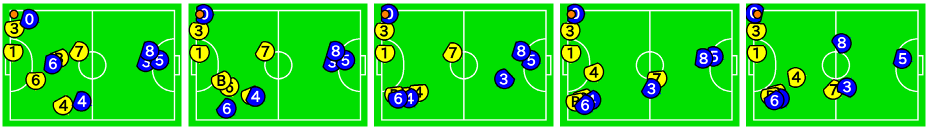


Fig. 5. Results obtained from classifying Skuba's set plays. (In the figure, the Blue team's robots are shown as black circles and the Yellow team's robots are shown in white. Figure 6 uses the same color convention.)



4 seconds before kicking | 3 seconds before kicking | 2 seconds before kicking | 1 second before kicking | the moment of kicking

Fig. 6. Positions of robots, every second in set play X_{37}

behavior, based on the classification results obtained from the N behaviors executed previously. As an example, consider the process of learning during a set play in which the game is restarted by placing the ball near the touch boundary.

A dissimilarity function d_3 is defined between the deployment choice X_j in C_i and the current deployment choice X as follows:

$$d_3(t, C_i) = \frac{1}{|C_i|} \sum_{X_j \in C_i} d_1(t, T_r^{(j)}, T_e^{(j)}). \tag{8}$$

This equation gives the mean value of the dissimilarity between X and X_j for each X_j in C_i. For all clusters obtained thus far, Equation (8) is computed to estimate the most likely deployment strategy selected by the opponent. The real-time computation is easy. Equation (8) used $T_r^{(j)}$ instead of $T_s^{(j)}$, as defined in Equation (5) because the action immediately prior to the kick as well as the precursors to the action taken in the strategy were of interest.

The data logged during the final game of the 2012 RoboCup competition were used to compute Eq. (8) for the 37th set play X_{37} of Skuba, assuming that the set plays $X_1 \ldots X_{36}$ could be classified as described in Section 5.2 (excluding X_{37} from C_3). Figure 7 shows the obtained results, and Figure 6 shows the positions of the robots every second during the set play X_{37}.

Because the value of $d_3(t, C_i)$ was high for C_1 and C_4 4 seconds prior to kicking, the strategy executed here was not selected from among these strategies. Clusters C_2 and C_3 had low values of $d_3(t, C_i)$ between 4 seconds and 2 seconds prior to kicking; however, the value for C_2 began to increase 2 seconds prior to kicking. Skuba's ID3 robot dashed out at this time. One second prior to kicking, the ID8 robot dashed out. Only C_3 had a low value of $d_3(t, C_i)$ after receiving the signal from

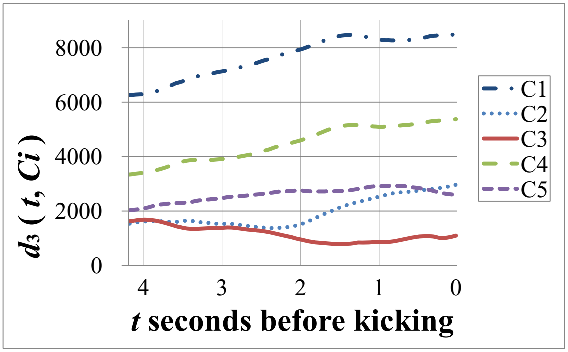


Fig. 7. The values obtained from Eq. (8). Duration: from the receipt of a start signal for a set play (from the referee box) to the completion of the kick.

the referee box. These observation, in conjunction with Eq. (8), suggested that the strategy corresponding to C_3 would be executed once again.

According to the strategy corresponding to C_3, the ball was passed to the robot that dashed out from the defense area in all cases. Therefore, in X_{37}, the ID8 robot was expected to shoot, and the ID3 robot, which dashed out 2 seconds prior to kicking, would be a bait robot. From this analysis, marking the ID8 robot 2 or 4 seconds prior to kicking will break the opponent's strategy.

7 Concluding Remarks

A dissimilarity function d that measures the differences between two different deployment choices in an SSL game has been designed. Additionally, a method for classifying and analyzing an opponent's strategies based on a clustering algorithm has been proposed. This method was applied to simulated games to demonstrate the utility of the method. Finally, the method was applied to the final game of the 2012 RoboCup SSL to show that an unknown strategy could be accurately classified. It has been demonstrated that the method could be used to an identify opponent's strategies during a game on the fly and in real time. Future work will further improve the method described here.

Acknowledgements. This work was supported by funding from the Aichi Prefectural University.

References

1. Panyapiang, T., Chaiso, K., Sukvichai, K., Lertariyasakchai, P.: Skuba 2012 Extended Team Description (2012), http://robocupssl.cpe.ku.ac.th/robocup2012:teams (accessed September 15, 2013)
2. Bowling, M., Browning, B., Veloso, M.M.: Plays as Effective Multiagent Plans Enabling Opponent-Adaptive Play Selection. In: Proceedings of International Conference on Automated Planning and Scheduling 2004, pp. 376–383 (2004)
3. Trevizan, F.W., Veloso, M.M.: Learning Opponent's Strategies In the RoboCup Small Size League. In: Proceedings of AAMAS 2010 Workshop on Agents in Realtime and Dynamic Environments (2010)
4. Visser, U., Weland, H.-G.: Using Online Learning to Analyze the Opponent's Behavior. In: Kaminka, G.A., Lima, P.U., Rojas, R. (eds.) RoboCup 2002. LNCS (LNAI), vol. 2752, pp. 78–93. Springer, Heidelberg (2003)
5. Davies, D.L., Bouldin, D.W.: A Cluster Separation Measure. IEEE Transactions on Pattern Analysis and Machine Intelligence 1(2), 224–227 (1979)
6. Sturges, H.A.: The Choice of a Class Interval. Journal of the American Statistical Association 21, 65–66 (1926)
7. SSL Technical Committee. Laws of the RoboCup Small Size League 2012 (2012), http://robocupssl.cpe.ku.ac.th/_media/rules:ssl-rules-2012.pdf (accessed September 15, 2013)
8. Everitt, B.S., Landau, S., Leese, M., Stahl, D.: Cluster Analysis, 5th edn. Wiley (2011)