UDC 004.89

Moriyama A., Mozgovoy M.

Classification and Clustering for Soccer Analytics

1 Introduction

Numerical analysis is often used in the field of sports. Analytical data is often shown in real time when sport events are broadcast live on television. Details of analysis are also posted online [1]. Sports analysis is useful for players, coaches, spectators, and television audience.

In the game of soccer, typical TV analysis shows statistical data, such as the number of free kicks, penalty kicks, offsides, the list of goal scorers, and players' total on-field time. Game overview and team strategy prediction is also often discussed.

However, the analysis of soccer games is not as developed as other sports [2], in part due to the overall complexity of soccer. In particular, there are 22 free-moving players in the field. So we have to follow all the players at the same time to analyze the situation. Consequently, a large number of calculations are needed. Another factor is the players individual abilities. These are as important in soccer as in other sport games (such as baseball), but they are more difficult to measure. For example, baseball players are usually engaged in well-defined actions such as throw, hit, catch, and run, so it is easier to measure their performance. A soccer player needs to possess a good combination of skills, which makes analysis challenging.

In this paper, we propose a method for classifying soccer game situations into separate clusters of similar situations. By developing this method, we expect to gain understanding of trends and strategies in typical and rare situations, occurred in the match. We tested and refined the algorithm using a real soccer dataset, containing games played by actual J.League teams.

Moriyama Akitaka – Graduate student, University of Aizu;
e-mail: sss1200110@gmail.com, phone: +81-242-37-2664

Mozgovoy Maxim – Associate Professor, University of Aizu;
e-mail: mozgovoy@u-aizu.ac.jp, phone: +81-242-37-2664

2 Methodology

2.1 Preprocessing

For testing our approach, we used a real soccer game dataset provided by the DataStadium company [3]. The dataset contains five games: Kobe versus Nagoya (9.7.2011), Nagoya versus Shimizu (7.5.2011), Shimizu versus Yamagata (15.6.2011), Urawa versus Yokohama (3.5.2011), and Yokohama versus Kobe (5.6.2011). The structure of each game in the dataset consists of three chunks: 1) current frame number; 2) each player data; 3) ball data. Each player is characterized with the following attributes: team ID, unique player ID, player jersey number, field coordinates, and moving speed. The attributes of the ball are its coordinates, speed, the ID of the owning team, and ball status. In this paper, we will show the results of processing the match between Kobe and Nagoya teams played on July 9, 2011.

On the dataset preprocessing stage we select only the game situations, satisfying the following conditions:

- Ball status is "alive".
- Number of players in the home team is eleven.

Sometimes soccer game are suspended. For example, when the ball is out of field bounds, the ball crosses the goal line, or the referee stops the game due to a foul. The teams do not make formations during the game pause until they resume play, so we only extract situations from the non-suspended fragments of the game. For the sake of simplicity and to reduce required computational resources we only follow the players of the home team, and extract only about 1000 uniformly distributed frames of the game.

2.2 Clustering Game Situations

For the purpose of the present research we decided to adopt a hierarchial clustering procedure, since it allows further analysis of individual cluster structure, which helps to make more fine-grained conclusions about analyzed games.

Hierarchical clustering algorithms are divided into two classes: divisive and agglomerative [4]. In this paper, we adopt a popular and simple agglomerative clustreing procedure. It is performed in a bottom-up way.

We set each game element object as a separate cluster, then we merge each pair of closest clusters recursively until we obtain a single cluster that includes all the elements of our dataset. Our algorithm consists of the following steps.

Step 1: calculate the distances between all the clusters in the dataset. We store the calculated distances in a distance matrix, and proceed to the Step 2.

Step 2: condition of termination. Since we use agglomerative clustreing, we should finish the procedure when a desired number of clusters (or the final single cluster that includes all the situations) is obtained. Otherwise we go to the Step 3.

Step 3: find a pair of a clusters to be merged together. By using the distance matrix we can search for the minimal distance between previously unused clusters. Next we proceed to the Step 4.

Step 4: merge the pair of clusters to make a new cluster. Since the obtained cluster needs a centroid, we set as a centroid of the combined pair the cluster containing more game situations. After this operation we return to the Step 2.

2.3 Calculating game situation similarity

Clustering algorithm needs a distance function for comparing soccer game situations. The problem of game situations similarity calculation can be regarded as an assignment problem, which is a classical task of mathematical programming and optimization [5].

In assignment problem, the task is to find the optimal correspondence between the elements of two sets. Each correspondence incur a certain cost, value, so the goal is to minimize the total cost. For example, suppose there are three workers and three jobs. One worker can take strictly one job, and there should be no role duplication. The time required to finish each job is different for different workers, so we need to minimize the total time.

In our case, the goal can be states ad the optimal assignment of player pairs across two different game situations to minimize the total Euclidian distance between the players (see Figure 1). We solve the assignment problem using Hungarian algorithm, also known as Kuhn-Munkres algorithm [5]. Its asymptotic complexity is $O(N^3)$.

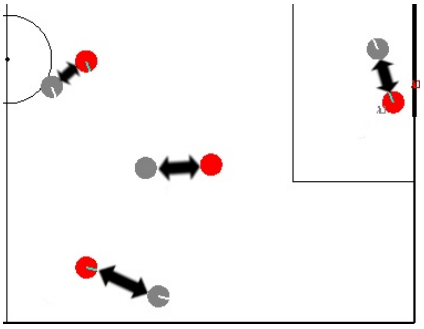

Figure 1. Correspondence between the players

3 Results

The figure 2 shows the histogram of the results of classification. We stopped the clustering algorithm when the number of clusters reached 100. The length of each bar is proportional to the cluster size (the number of game situations contained in the cluster).

Similarity between game situations decrease rapidly with each new cluster. The largest 14 clusters include about 64% of all game situations. The Figure 3 shows four typical soccer game situations, taken from the two largest clusters identified. Similarly, the Figure 4. shows four rare game situations obtained from the two smallest clusters. The home team is on the right side of the field, the away team is on the left side.

4 Discussion

Cluster analysis helps to reveal typical and rare situations in the game of soccer. For example, in the cluster 1 shown in the figure 3, we see that the ball and almost all the players of the home team are present on the away team's side of the field, forming a typical attack pattern. In the cluster 2, the home team makes a formation consisting of vertical lines. The majority of away team's players are located on the home team's side, so the home team makes the formation to defend against the attack of the away team.

In the cluster 1 of the figure 4 both teams are located near the field center. The distances between the players is small, so the players scram-

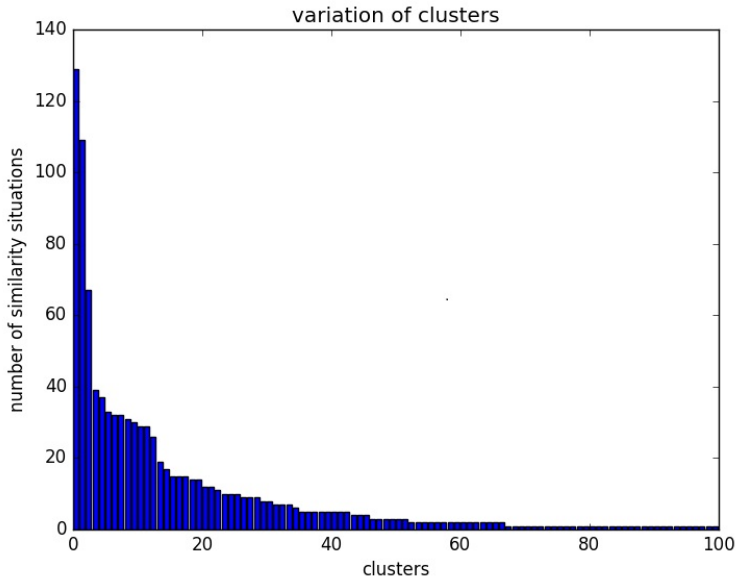

Figure 2. Distribution of game situations in clusters

ble for the best position to get the ball. According to the soccer game video, this is a moment which the game resumed play after a pause. In the cluster 2, the home team defend an attack of the away team. The players make a formation of a vertical line. This is the moment when the game was resumed after a free kick.

It is evident from the results that different clusters correspond to distinct game situations where team choose different player formations. We can observe situations occurring frequently in large clusters. For example, the players are frequently positioned in the middle of the field, and the home team attacks/defends like shown in the cluster 1 and cluster 2 of the figure 3. Small clusters consisting of rare game situations often include positions, occurring when the game is resumed after pause.

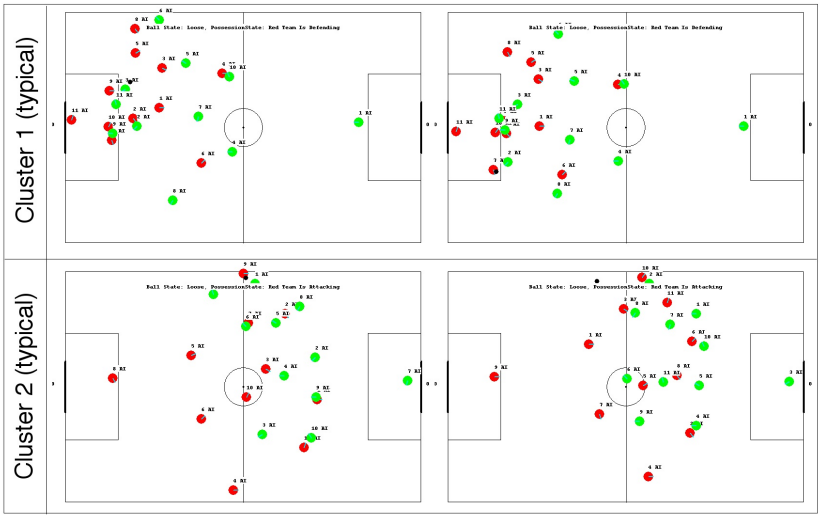

Figure 3. Clusters with typical game situations

5 Conclusion

We classified soccer game situations into typical and rare situations using hierarchical clustering. In the present work we defined game situation distance as the sum of all Euclidean distances between pairs of assigned players. We used Hungarian algorithm to find the optimal assignment of the players. We hope that this method will contribute to the methods of deeper analysis of soccer games, since it helps to understand the nature of soccer and identify frequent and rare game situations.

References

1. J.League: Japan Professional Football League. `www.jleague.jp` (Accessed: 28.01.2016).

2. I. Miki. Do soccer games cut out for data analysis? (in Japanese). AtMarkIT, 29 September 2015 [Internet resource]. `www.atmarkit.co.jp/ait/articles/1509/29/news029.html` (Accessed: 25.01.2016).

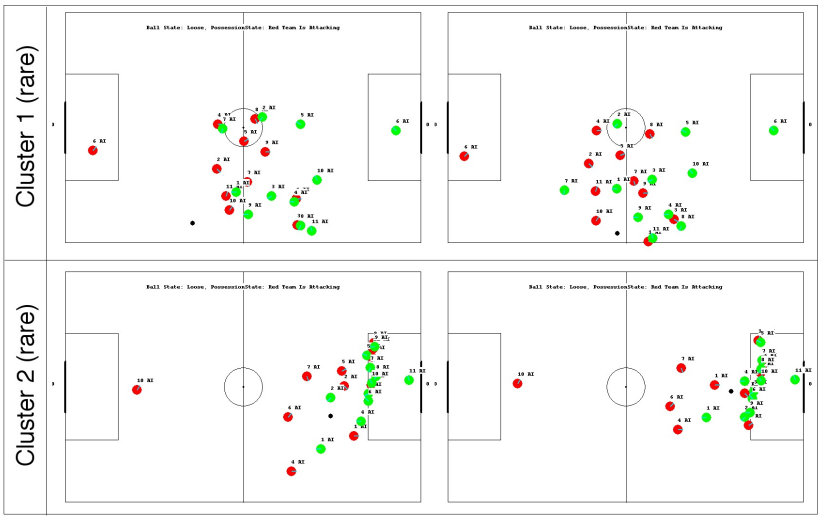

Figure 4. Clusters with rare game situations

3. Data Stadium Inc.
`www.datastadium.co.jp` (Accessed: 31.01.2016).

4. A.K. Jain and R.C. Dubes. Algorithms for clustering data. Prentice-Hall, 1988, 304 P.

5. J. Munkres, Algorithms for the assignment and transportation problems // Journal of the Society for Industrial and Applied Mathematics. 1957. Vol.5, No.1, P.32-38.