Similar Situations Identification for the Game of Soccer

Georgii Mola Bogdan Maxim Mozgovoy

Active Knowledge Engineering Lab, University of Aizu
Itsukimachi Oaza Tsuruga,
Aizuwakamatsu, 965-8580 Japan

E-mail: {d8212101, mozgovoy}@u-aizu.ac.jp

Abstract

In this work, we study the repeatability of game situations in different soccer matches. This analysis is aimed to evaluate the possibility of reusing these records as part of a game AI system. Due to a variety of team formations, an appropriate comparison of game situation pairs is a challenging task. Identification of similar situations in the game of soccer can be presented as an evaluation of geometrical similarity of players' coordinates on the field. Team formations have semantic value, and we show that role-based analysis is essential for successful matching. Obtained results can be applied for the tasks of sports analytics and AI design.

1 Introduction

In our time sports analytics has turned into a huge industry that aggregates more and more technologies from engineering and computer science.

Modern video recording hardware allows to obtain high quality scenes of any sport match. At the same time, advanced software powered with algorithms of image recognition opens a wide spectrum of possibilities for sport teams, sport analysts, spectators, and researches. In particular, it can be used by video game AI developers.

Data providers (such as Data Stadium Inc. [1]) process video streams of soccer matches and convert them into digital form. Originally, such data served to perform various statistical and visual analytics, performance evaluation, and tactics improvement. It was also used by broadcasting companies to enhance spectator experience. Currently, real sports tracking data is becoming more and more available to the general public, which allows to ask whether it is possible to exploit such data for extraction of behavior patterns of real teams. In its turn, result of such processing can be used for development of a virtual team that preserves real players behavior.

One of possible approaches to design a virtual team is Case-Based Reasoning (CBR) [2]. It served as a basis for several works related to RoboCup competitions [3-5], where *virtual* AI teams demonstrated the ability to behave coordinated and effectively. Does that efficient in comparison with humans play? Michael and Obst [6] note that teams of humans *"were easily won by computer programs"*, adding though that *"the soccer simulation was not designed to be played by humans"*. However, video game AI is not required to provide ultimate efficiency, Its typical goal is to be reasonably strong and provide human-like behavior that is often considered as prerequisite for fun [7].

Basically, soccer is game of spatial tactics; therefore, individual observable cases can be represented by a set of geometrical data like players coordinates and ball coordinates. Thus, the task of searching for similar cases can be treated as search in multidimensional space (which, in its turn, is based on the assumption that situations do repeat).

As a preliminary step for this work, we decided to evaluate the repeatability of soccer matches using a relatively small dataset.

2 Data set

For this work, we used digitalized soccer recordings obtained with TRACAB technology [8] that relies on video streams obtained from six still video cameras installed in a stadium.

The resulting data files were gathered with a frame rate of 25 frames per second. They contain various

values describing game states and consist of colon-separated chunks, representing different values.

The dataset used in this work is collected and provided by Data Stadium Inc. [1]. It consists of 5 games of 6 teams. The game data originates from J1 League matches (Japanese top division soccer league). All games were played in the 2011 season, and conventional statistical data (including team formations) is available online.

3 Formation analysis

One of the significant challenge factors in our task is small size of the used data set, comparing to a typical data set size sufficient for most conventional machine learning algorithms. Hence, the repeatability test itself served as proof of concept for using CBR with a limited dataset.

In the process of repeatability evaluation, it is important to keep an appropriate player ordering. Such a measure helps to increases a chance to find a matching case with a similar meaning. If wrong players will be paired, false-negative results will appear during search, and discovered cases may not be useful. (The process of ordering players in accordance with their roles is called "role-alignment" in [9]).

We started with the calculation of centroids of all 5 matches and identified root mean square error (RMS) for individual players using the formula

$$\hat{\sigma} = \sqrt{\frac{1}{n}\sum_i((x_i - \bar{x})^2 - (y_i - \bar{y})^2)},$$

where (x_i, y_i), i=1, 2, …, N are the point coordinates, and the corresponding centroid (point of averages) is $(\bar{x}, \bar{y})$. RMS is used to show whether positions of players significantly differ from the average positions. The obtained results are shown in the Table 1.

Table 1. RMS of players positions in 5 games.

Player №	RMS by games, m					Overall RMS, m
	Game number					
	1	2	3	4	5	
1	7.04	10.02	7.54	8.04	9.56	2.92
2	22.42	20.73	22.92	25.1	24.48	6.62
3	18.55	20.75	21.43	26.68	22.06	9.33
4	19.11	25.66	23.41	28	22.88	6.2
5	23.28	21.32	25.08	27.21	25.85	12.53
6	23.56	21.54	27.51	24.18	21.64	9.57
7	23.19	22.09	21.75	26.41	26.93	5.94
8	20.7	24.43	25	29.89	28.38	9.01
9	27.35	22.77	26.97	27.5	26.39	8.87
10	26.63	21.66	23.61	28.54	23.48	6.21
11	22.23	21.41	21.96	30.39	24.8	7.5
12	7.16	8.55	8.63	8.83	6.47	0.99
13	26.02	22.11	19.6	23.98	22.94	12.02
14	23.08	19.52	25.01	24.24	20.56	9.31
15	23.54	21.91	21.94	25.37	20.07	9.18
16	21.29	27.62	19.72	22.15	23.33	12.18
17	24.3	19.02	26.11	29.95	21.76	10.39
18	29.28	19.99	24	27.39	28.03	12.21
19	25.22	26.59	25.68	29.03	23.03	10.02
20	23.41	25.01	23.65	23.76	24.14	9.18
21	24.7	24.9	23.88	29.35	26.15	6.53
22	21.9	25.14	22.94	26.85	25.85	7.19

Every row corresponds to the position of a player in a single data set record. According to the Table 1, the players 1 and 12 have the most limited area for their movements. In our dataset, these players are always the goalkeepers. The rest of the players have similar distance values, which allows to expect relatively stable position patterns in matches.

4 Optimal matching with Hungarian algorithm

Our experimental approach suggests to search a game situation of one game inside one of the remaining 4 games. Such procedure is repeated for every game.

In order to get best player-player matching, we applied Hungarian algorithm [10]. It takes as an input a 22×22 cost matrix, where every row contains the distance between a player from the game situation S and all the players from the situation S'. The output of this algorithm is an optimal assignment for the original cost matrix. While the complexity of this algorithm is too high for real-time processing, it can still provide a reference for estimating the expected number of matching cases.

Game situation comparison in soccer requires correct players matching. For example, goalkeeper of own team should be compared with goalkeeper that considered to be "own" in each extracted game situation. It is also necessary to avoid pairing players from different teams. Conventional Hungarian algorithm does not take into account players team affiliation, so we built the distance matrix (1) in the following manner: the distances between different team members are substituted with the "infinity" value.

$$\begin{pmatrix} \infty_{1,1} & \cdots & \infty_{1,11} & D_{1,12} & \cdots & D_{1,22} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \infty_{11,1} & \cdots & \infty_{11,11} & D_{11,12} & \cdots & D_{11,22} \\ D_{12,1} & \cdots & D_{12,11} & \infty_{12,12} & \cdots & \infty_{12,22} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ D_{22,1} & \cdots & D_{22,11} & \infty_{22,12} & \cdots & \infty_{22,22} \end{pmatrix} \quad (1)$$

Indexes from 1 to 11 correspond to one team and 12

to 22 to another team respectively.

So, the algorithm will never assign a player from the team A to a player from the team B even if distance between them is minimal.

The obtained result of every comparison used to recompose given game situation and get optimal matching. After the optimal player/player pairing is identified, we need to choose the measure of game situation similarity. Since every soccer situation consists of floating-point numbers (coordinates), searching for exact matching is unreliable and often unnecessary, because game situations change continuously. Furthermore, precise matching requires larger datasets, but they are hard to prepare for soccer, and thus unlikely to appear in near future. That is why the comparison procedure simply considers two points matching if the Euclidean distance between them is smaller than the given range value. The results of similar cases identification are shown in Table 2.

Table 2. Search with Hungarian algorithm.

Success rate in matches, %					
Range,	Game number				
m	1	2	3	4	5
4	0.37	0.41	0.42	0.43	0.24
5	6.93	7.99	6.54	7.09	7.29
6	18.07	19.78	17.20	18.28	18.84
7	22.75	23.62	22.19	22.73	23.05
8	24.20	24.70	23.77	24.09	24.05
9	24.65	24.95	24.21	24.7	24.4

Where *success rate* indicates how many game situations in "knowledge base" matched (within given range) with incoming game situation. By specifying shorter ranges, we can find closer matches, at the higher risk of retrieving no results at all.

5 Case extraction based on linear search

Due to relatively small size of data set it was possible to perform experiments based on naive search algorithms. First, we performed a straightforward linear search with the original data set. The obtained results are shown in Table 3.

Table 3. Linear search.

Range,	Success rate in matches, %				
m	G1	G2	G3	G4	G5
7	0.01	0.01	0	0	0
8	0.11	0.06	0.22	0.06	0.16
9	0.69	0.46	0.74	0.19	1.01

Comparing with Hungarian algorithm that provides optimal pairing for all processed situations, this approach produces results that are insufficient for practical use. Another experiment was based on the same linear search, but the data set was constructed using centroids and Hungarian algorithm. We used the following procedure: before adding record into data set, it was compared with the centroids obtained during processing data for Table 1, representing a certain "average formation". The approach was used for recombination of records from the test set. Hence, every record from test set preprocessed to have "proper" order in according to reference formation. It is important to note that centroids from search data set were used for the test set due to two reasons: 1) the test set imitates incoming situations unknown in advance, and 2) measured deviation allows to assume that all games have similar average formations. Thus, all players in every game situation are arranged optimally for the position comparison procedure. The following table 4 shows the results.

Table 4. Linear search (centroids-based data set).

Range,	Success rate in matches, %				
m	G1	G2	G3	G4	G5
6	0.79	1.48	1.33	1.18	1.40
7	4.65	6.25	5.38	5.27	6.31
8	11.79	13.14	11.35	11.22	12.92
9	18.00	18.79	16.58	16.23	18.43

The obtained results have significantly higher success rate. However, the complexity of both approaches is $O(N)$ (we do not take into account preprocessing for the 2nd method), which is sufficient for evaluation purposes. However, in actual soccer AI system a faster method is necessary.

6 Case extraction based on kd tree

As soccer is fundamentally a game of spatial tactics, one would expect to deal with high-dimensional geometric data like team members and ball. That is why we consider applying kd tree algorithm as a relatively simple, straightforward and effective way that allows to retrieve close points in a multidimensional space. Algorithms based on kd trees are widely used in the domain of computer graphic, for instance, in tasks like ray tracing or color reduction; however, they can also be applied for case-based reasoning tasks, as suggested in [11]. This method possesses a number of attractive features: it gives us an explicit criterion of similarity between the current onscreen game situation and game situations in the training dataset; it is computationally inexpensive; it allows us to see specific base cases for each decision, thus helping to fine tune and improve the system.

As was discussed in Section 4, the search for an exact value is not a reliable approach. Thus, the essential feature of a kd tree is search within a specified range

in multidimensional data. The kd tree-based method is fits for the task of searching closest matches for complete 22-element vectors, containing the coordinates of all soccer players. As a result, a kd tree contains a set of 44-dimensional points. The construction complexity for a kd tree is $O(N \log N)$, and memory consumption is $O(N)$.

Querying an axis-parallel range in a balanced kd tree has the complexity of $O(N^{1-\frac{1}{k}} + m)$, where m is the number of reported points, and k is the dimension of the kd tree (in our case, k=44) [12].

The overall process of data processing presented in figure 1.

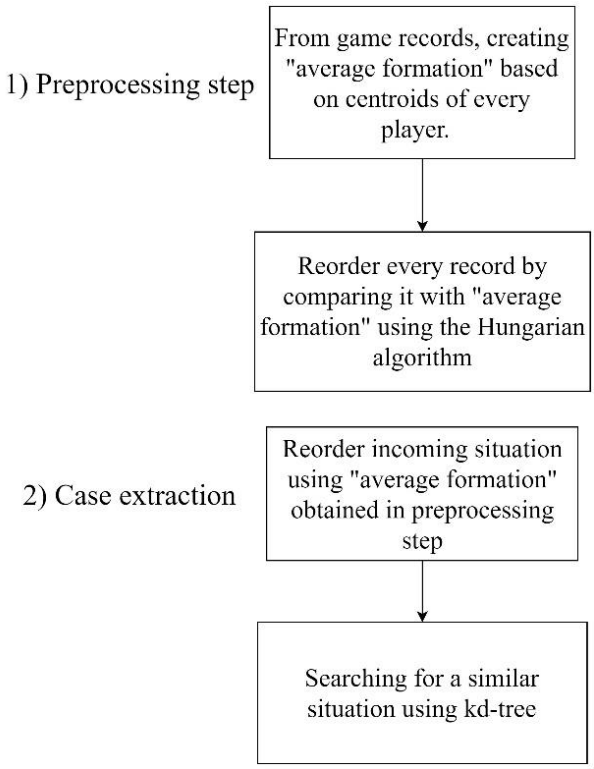

Fig. 1. Workflow diagram.

7 Conclusions

We evaluated repeatability of soccer game situations and influence of role-alignment on search results. Current results can serve as proof of concept for using real soccer digitalized data as a basis for a CBR system. Also, we showed how even trivial formation analysis (centroids) can enhance the rate of case extraction. In Table 3 the success rate for 7-9 meters is close to zero while using centroids and Hungarian algorithm (Table 4) increases success rate in several times for the same ranges and provides results for 6 meters.

Further steps will require to filter results of queries thus there are can be less actual effective results. It caused by subtle features included in every game situation that can significantly change semantic value. However, taking into account opportunity to increase data set, we consider such results as promising.

The current approach can be useful for other multi-agent team games where geometrical composition plays an important role.

References

[1] "Data stadium Inc." https://www.datastadium.co.jp, accessed: 2019-10-20.

[2] A. Aamodt and E. Plaza: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches, *Artificial Intelligence Communications 7*, pp. 39-59, 1994.

[3] T. Steffens: Similarity-based Opponent Modelling using Imperfect-Domain Theories, *CIG'05*, pp. 285-291, 2005.

[4] R. Ros, M. Veloso, R. L. de Mantaras, C. Sierra, J. L. Arcos: Retrieving and Reusing Game Plays for Robot Soccer, *ECCBR'06: Advances in Case-Based Reasoning*, pp. 47-61, 2006.

[5] T. P. D. Homem, D. H. Perico, P. E. Santos, R. A. C. Bianchi, Ramon L. de Mantaras: Qualitative Case-Based Reasoning for Humanoid Robot Soccer: A New Retrieval and Reuse Algorithm, *ICCBR 2016: Case-Based Reasoning Research and Development*, pp. 170-185, 2016.

[6] O. Michael and O. Obst: BetaRun Soccer Simulation League Team: Variety, Complexity, and Learning, *arXiv preprint arXiv:1703.04115*, 2017.

[7] M. Sicart: A tale of two games: football and FIFA 12, *Sports Videogames*: *Routledge*, pp. 40–57, 2013.

[8] "Tracab," http://chyronhego.com/sports-data/tracab, accessed: 2019-10-20.

[9] M. Mozgovoy and I. Umarov: Believable team behavior: Towards behavior capture AI for the game of soccer, *8th International Conference on Complex Systems*, pp. 1554–1564, 2011.

[10] H. W. Kuhn: The Hungarian Method for the assignment problem, *Naval Research Logistics Quarterly*, 83-97, 1955.

[11] H. M. Le, Y. Yue, P. Carr, and P. Lucey: Coordinated multi-agent imitation learning, *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1995–2003, 2017.

[12] J. L. Bentley, J. H. Friedman: Data structures for range searching, *ACM Computing Surveys (CSUR), vol. 11*, no. 4, pp. 397–409, 1979.