Towards Case-based Reasoning with k-d Trees for a Computer Game of Soccer

Georgii Mola Bogdan

Maxim Mozgovoy

Active Knowledge Engineering Lab
University of Aizu
Aizuwakamatsu, Japan
{ m5212104, mozgovoy}@u-aizu.ac.jp

***Abstract*—Real soccer tracking data can potentially be used for building a learning by observation-based game AI system, possessing human-like decision making traits. Such a system should be able to find matching example cases for the current game situation and act accordingly. This can be a challenging issue due to high dimensionality of soccer data and the requirements to detect approximate matches. We show how a simple k-d tree-based search can accomplish this task with modest space and time requirements, making it a feasible approach for a practical game system.**

Keywords—soccer, case-based reasoning, k-d tree, game AI

I. Introduction

Creating a reasonable AI system for the game of soccer is a challenging task, addressed in numerous research works. Designing high-performance soccer teams consisting of physical robots or virtual agents is the goal of annual RoboCup competition [1], attracting numerous participants and observers. Currently, most successful 2D Simulation League teams employ diverse techniques to obtain efficient, winning behavior [2].

Advancements in machine learning methods motivate some teams to experiments with approaches based, e.g., on neural networks. One such example is Brainstormers team [3] that won several major RoboCup events in 2000s. Machine learning can be used within the context of learning by observation methods, aiming to acquire patterns of successful behavior from the top teams. This idea is expressed, in particular, in the work of Michael and Obst [4], who propose learning from the best participants of the RoboCup 2D simulation league.

The growing availability of real sports tracking data (such as provided by Data Stadium Inc.) begs the question whether it is possible to learn *virtual* team behavior from *real* soccer teams. Michael and Obst [4] 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"*. Thus, it is possible (but not certain) that learning from human data would not help creating strong virtual teams.

However, being strong is not the only requirement for game AI systems. In soccer, it is important to provide a realistic and enjoyable environment for human players, facilitating immersion and suspension of disbelief. As Sicart [5] observes, the aim of mainstream soccer game projects is *"to make the games even closer to the actual game, that is, to make the computer game converge with the sport"*. In particular, it requires the AI systems to behave in "human-like" way or even mimic behavior styles of particular real-life soccer teams, which can be a strong motivation for learning from player tracking data. While it is not entirely clear what exactly constitutes "human-like behavior" in this context, it is known that individual professional soccer teams exhibit identifiable behavioral patterns [6, 7].

Learning soccer strategies from observation is a challenging task, since agent coordination is implicit, and dimensionality is very high [8]. However, there are recent promising results, obtained primarily with deep learning technologies [8, 9].

Since computer soccer is essentially a game of spatial tactics, existing learning by observation approaches are primarily based on spatial features, such as player and ball coordinates. This observation motivated us to experiment with a straightforward k-d tree-based approach, able to find close points in a multidimensional space, as suggested in [10]. 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. In the present paper, we are primarily addressing the problem of case retrieval. However, the eventual goal is to support real-time decision making as well.

II. Goals and Methods

At the current stage of the work, we aim to investigate how easy and computationally demanding is to find a base case in the training data for the given onscreen situation. In one of our previous projects [11], we tried to base decision making on players' local contexts without relying on the complete team data, which greatly reduces problem dimensionality.

One of the key reason to use k-d tree is the ability to perform search within a specified range in multidimensional data. Since we are dealing with floating-point numbers (coordinates), searching for exact cases is unreliable. The k-d tree-based method is fast enough to search closest matches for complete 23-element vectors, containing the coordinates of all soccer players and the ball (46 values in total). These vectors should list players consistently in the same order according to their roles, so initial data preprocessing is inevitable (this step is called "role-alignment" in [9]). If role alignment is performed incorrectly (for example, a defender is paired with a forward player), additional false negative results will reduce the success rate of algorithm. Since any role-alignment algorithm is heuristic in nature, and its quality must be taking into account as a part of evaluation of obtained results, we decided to base our preliminary experiments on a smaller tracking dataset, where the roles of all players are known. Thus, no automated role-alignment is necessary.

Our current dataset consists of five complete matches, where the opponents are six different teams of the top J1 Japanese soccer league. All games were played in the 2011 season, and conventional statistical data (including team formations) is available. Each game is represented as a sequence of frames, taken at the rate of 25 frames per second. Each frame, in particular, includes two-dimensional player and ball coordinates. The players are sorted according to their unique IDs, which allows us to track individual player movements.

Since the original framerate is too high for the purposes of case extraction, we reduced it to one frame per three seconds. Thus, our resulting training set (4 matches) consists of ≈7500 frames, and the test set (1 match) contains ≈1800 frames. We should note that in this task being able to learn from limited dataset can be an important additional goal. If we want to obtain a virtual team that behaves like a specific real-life team, we should learn from the tracking data of this team, which would probably contain just a few matches (until player tracking technologies are universally adopted).

Adding new elements to a balanced k-d tree takes $O(\log_2 n)$ time, where n is the number of elements in the tree. In our case, each element is a vector of 46 floating-point numbers (two coordinates per each game object).

Querying an axis-parallel range in a balanced kd-tree has the complexity of $O(n^{1-1/k} + m)$, where m is the number of reported points, and k is the dimension of the k-d tree (in our case, k=46) [12]. To increase search speed, we retrieve at most 10 matching cases. In practice it keeps search time under 1ms on a conventional Intel i7-based laptop PC for our dataset.

We have to mention that k-d trees are unable to provide any long-term planning mechanism. This drawback is probably not crucial for the game of soccer, but might cause issues in other game genres. It is also important to note that geometrical similarity is not a reliable indicator of "true" semantic closeness between two individual game situations). Thus, case extraction with a k-d tree should be considered a fast search procedure for potential candidate cases, followed by subsequent deeper analysis.

III. EXPERIMENTS AND DISCUSSION

A range search procedure considers two points matching if the Euclidean distance between them is smaller than the given range value. By specifying shorter ranges, we can find closer matches, at the higher risk of retrieving no results at all.

Finding optimal range values is not a trivial task. In general, case extraction in soccer is affected both with the local context (especially in the area around the ball) and the global situation on the field. The global context helps to identify "the big picture": whether a certain team is attacking or defending, how close the ball is to the opposing team's goal line, or whether the attacking team is about to play a set piece. This context can be identified with a relatively low-resolution matching (i.e., with higher range values). The local context determines specific actions of individual players, and thus requires much higher precision around the ball area. For example, matching defenders' positions precisely is vital for a successful attack, while the location of own goalkeeper is nearly irrelevant.

Therefore, it is reasonable to test the outcomes of k-d tree queries both for the whole soccer field and for the limited area within the given radius around the ball. In practice, we are planning to combine local and global queries in the resulting decision making system: a low-resolution global query will be performed to identify the general context, and the subsequent precise local query will be used to find the best match for the given player arrangement.

Global (whole field) queries for our test set yield the probability of finding a match within 0.5-1.0% for the range of 8 meters, and it reaches 50% for the ranges of 12 meters and above. The results of local context queries are shown in the Table 1.

TABLE I. SUCCESSFUL LOCAL CONTEXT QUERIES (%)

Range, m	**Radius around the ball, m**					
	4	**5**	**6**	**7**	**8**	**9**
1	37.4	27.5	21.2	17.8	15.6	13.5
2	55.0	42.0	32.1	25.2	20.1	18.0
3	70.3	56.0	43.7	33.7	27.3	22.7
4	82.0	69.9	56.5	46.0	37.2	29.9

For proper evaluation of these results, additional work on actual decision-making subsystem is necessary. However, we consider them promising. For the strictest local context query (range = 1m, radius = 9m), there is a 13.5% chance to find a base case for decision making. If no results are found, we can repeat the query with relaxed conditions by providing higher range and/or smaller radius values until some case is found.

To analyze practical examples of identified matches, we visualized results. Fig. 1 shows overlapped images of a query situation and the corresponding retrieved matching result for a global search procedure (range = 8m). Fig. 2 shows an example result of a local query (range = 1m, radius = 9m). In both figures, players painted with a solid fill, belong to the query situation, while semitransparent-filled players represent a match found in the k-d tree.

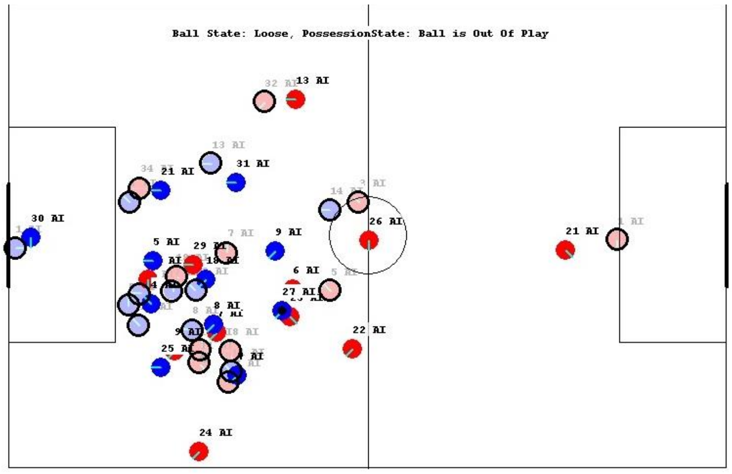

Fig. 1. Situations matched as a result of a global query (range = 8m).

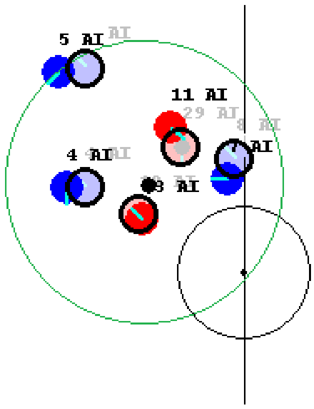

Fig. 2. Situations matched as a result of a local query (range = 1m, radius = 9m).

IV. Conclusion

We have applied a k-d tree-based approach for case retrieval in a soccer game. Our primary goal at this stage was to show the feasibility of this method both in terms of computational performance and its ability to find matching cases in the database. While our current dataset is too small to be used as a reliable source for decision making, its content is already sufficient for local context analysis of the field area surrounding the ball. Global reasoning is also possible: while the probability of finding a match with sufficient precision is not high, global context does not change rapidly, which makes frequent decision making unnecessary.

Apart from experimenting with larger datasets, role alignment, and actual decision making, we are planning to test the hybrid global/local approach, where an admissible range would increase for soccer field areas, located far away from the ball. The number of identified local matches can be increased by focusing on relative locations of the players around the ball rather than absolute coordinates on the field, which is done in the present version.

References

[1] H. Kitano *et al.*, “RoboCup: A Challenge Problem for AI and Robotics,” *Lecture Notes in Computer Science*, vol. 1395, pp. 1–19, 1998.

[2] M. Prokopenko and P. Wang, “Disruptive innovations in RoboCup 2D soccer simulation league: from Cyberoos’ 98 to Gliders2016,” in *Robot World Cup*, 2016, pp. 529–541.

[3] M. Riedmiller, T. Gabel, F. Trost, and T. Schwegmann, “Brainstormers 2D — Team Description 2008,” *RoboCup 2008: Robot Soccer World Cup XII*, 2008.

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

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

[6] T. Decroos, J. van Haaren, and J. Davis, “Automatic Discovery of Tactics in Spatio-Temporal Soccer Match Data,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 223–232.

[7] L. Gyarmati, H. Kwak, and P. Rodriguez, “Searching for a Unique Style in Soccer,” Sep. 2014. [Online] Available: http://arxiv.org/pdf/1409.0308v1.

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

[9] H. M. Le, P. Carr, Y. Yue, and P. Lucey, “Data-driven ghosting using deep imitation learning,” in *Proceeding of the 11th MIT Sloan Sports Analytics Conference*, 2017.

[10] S. Wess, K.-D. Althoff, and G. Derwand, “Using k-d trees to improve the retrieval step in case-based reasoning,” in *European Workshop on Case-Based Reasoning*, 1993, pp. 167–181.

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

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