Braitenberg Soccer

Tekin Meriçli, H. Levent Akın {TEKIN.MERICLI,AKIN}@BOUN.EDU.TR
Derpartment of Computer Engineering, Boğaziçi University

Abstract

Well-developed individual and collaborative skills, such as dribbling the ball, positioning, and passing are required for a team of robots to be successful against an opponent team in a robot soccer scenario. This paper proposes an approach to individual and collaborative skill learning, where the robots are modeled as Braitenberg vehicles, and the required skills are implemented as combinations of very primitive behaviors. Without explicit communication and role assignment mechanisms, the robots were able to learn how to play soccer as a team after a short training session, where reinforcement learning was used to construct the optimal state-action mapping. Experiments demonstrate that a team of robots can indeed learn to play soccer reasonably well without using complex environment models and state representations.

1. Introduction

Multi-agent Systems (MAS) is the subfield of artificial intelligence (AI) that aims to provide both principles for construction of complex systems involving multiple agents and mechanisms for coordination of individual agents' behaviors (Stone & Veloso, 2000). Being a complex and dynamic environment, and having a goal that can be achieved more successfully with multiple agents than a single agent, soccer is an excellent testbed for MAS research.

Beating the opponent in a soccer game requires having well-developed individual skills; such as dribbling and kicking the ball, and collaborative skills; such as passing and proper positioning. However, none of these skills need to be perfect in order to play reasonably well. This observation leads to the the main motivation behind this work; that is, to create a team of autonomous robots that are able to play soccer by using combinations of very primitive behaviors, which was inspired from the complexity of behaviors that Braitenberg vehicles (Braitenberg, 1984) can demonstrate although their underlying architectures are extremely simple.

Because of the inherent complexity of MAS, Machine Learning (ML) is an interesting and promising area to combine with MAS in order to learn effective individual and collaborative skills. However, most of the ML applications require a large amount of labeled examples; that is, one has to provide information about thousands of different situations in order to make a machine learn a concept. On the other hand, in robot soccer case, it is impossible to provide labeled examples to the system because of the complexity and dynamic structure of the environment. Therefore, a trial/error and reward/punishment approach is necessary to be able make learning possible in this domain.

Reinforcement Learning (RL) is a learning method that can be used when the agent is only informed about the degree of correctness (or incorrectness) of a sequence of actions. Specifically $Q(\lambda)$ algorithm (Sutton & Barto, 1998), which is a RL method, is utilized in this work. Experimental results show a significant decrease in the number of opponent goals, which means that the team learned a defensive behavior, as well as an increase in the number of own goals, which is an indicator of a learned offensive behavior.

The rest of this paper is organized as follows. Section 2 elaborates on our proposed approach. Experiments are explained in detail and results are discussed in Section 3. Section 4 summarizes and concludes the paper, and proposes some further extensions.

2. Approach

The main motivation behind this work is that playing soccer reasonably well does not require highly complicated models for each skill, vast state spaces, and long training sessions that last for thousands of episodes; at least for non-robot soccer players. Hence, we aimed keeping the representation simple yet informative enough in order to make it possible for our robot soccer players to learn individual and collaborative skills. Taking a closer look at the general structures of these skills, we see that they all have two primitive behaviors in common: *moving towards a point* and *moving away from a point*. These behaviors can easily be implemented as the behaviors of Braitenberg vehicles; such as "aggression" and "fear", and the implementation can be approximated using force fields as defined in (Kaplan & Akin, 2003). An attractive force field is placed on the ball as well as a circular field that makes it possible for the robot to end up facing towards a target point when it meets the ball. Other robots and the border lines have repulsive force fields on them in order to keep our robot away from them and inside the border lines, respectively.

Interpretation of these behaviors, illustrations of which are provided in Figure 1, is as follows.

- *moves towards* the ball **and** *moves towards* the opponent goal, ***attacker*** behavior is observed
- *moves towards* the ball **and** *moves towards* the home goal, ***defender*** behavior is observed
- *moves away from* the ball **and** *moves towards* the opponent goal, ***supporter*** behavior is observed
- *moves towards* the ball **and** *moves towards* a teammate, ***passing*** behavior is observed.

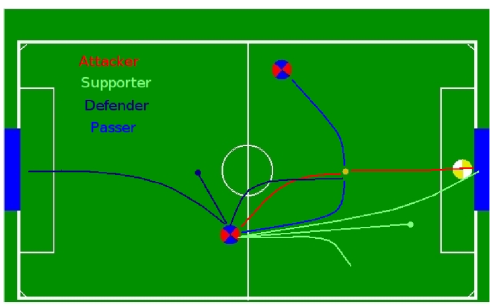

Figure 1. Combination of the primitive behaviors *moving towards a point* and *moving away from a point* to create more "complex" behaviors, such as offense (red), defense (dark blue), support (green), and passing (blue).

By discretizing the environment, we can represent the positions of each robot and the ball using the corresponding *cell ID*. The state of the immediate surrounding of the ball is defined as the *dominance value*, which is computed as the difference between the number of own robots and the number of opponent robots in the corresponding cell. Whether one of our robots, a teammate, or an opponent player is closest to the ball is another important environmental information. Figure 2 illustrates the discretization of the field and the dominance in a given cell, where the blue-red robots belong to our team and the dominance value inside the light-green colored cell is 0.

In a given state, the robot can perform one of the five different actions; namely attacking the opponent goal, supporting the attacker, defending the home goal, passing to the closest teammate, and passing to the teammate that is

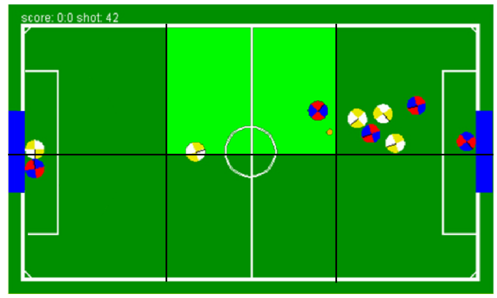

Figure 2. Quantized robot soccer field.

closest to the opponent goal. Therefore, the state-action mapping representation is as follows.

$$ballCell, closest, dominance, action \rightarrow probability$$

Initially the probability values are all equal and 0.2, which means that all five actions have equal chances to be selected in all states. Adjustment of those probability values is done through learning, in particular *reinforcement learning*. Specifically, the $Q(\lambda)$ (Sutton & Barto, 1998) algorithm is used in this work. After the Q values are computed, they are normalized so that they could be treated as probability values and the sum of the probabilities of possible individual actions in a given state would be 1.0. A delayed reward or punishment is given after a goal is scored and the last N actions are affected from this assignment inversely proportional to their temporal distances from the last action since these series of actions resulted in scoring or receiving a goal.

Since it is almost impossible for any two robots on the field to have the exact same distance to the ball, most of the time each robot in the team has a different state-action representation tuple for the same state. Therefore, at each time step, each robot "experiences" a different situation, makes a decision based on what it observes, and modifies a different part of the probability table. This experience table is shared among the teammates; that is, the robots implicitly share their experiences. What actually happens is that they communicate through changing the environment, which is very similar to what swarms do in nature (stigmergy). That eliminates the need for an explicit communication protocol, simplifying the model even further.

3. Experiments

The experiments were run on the TeamBots simulation environment (Balch, 2000), which can be thought of as a simulation of the FIRA - MiroSot league (FIRA, 2010), and an approximation of the RoboCup Small Size League (RoboCup Small Size League, 2010). A team of five players was considered and only the non-goalie players were trained. The goalie ran a very simple positioning code which placed the robot on the intersection of line that connects the ball and the center of the goal box and the goal line.

There are five different hyper-parameters that affect the course of the game which are the constant determining the exploration / exploitation rate K, the dimensions of the grid GD, the length of the state-action history HS, and the values of immediate I and delayed D rewards / punishments.

An incremental method is followed for training, where the team was first trained on an empty field, and then against opponents with gradually increasing strengths. At the end of each game, the *score-rate* (Equation 1), which is defined in terms of the difference between the own score and the opponent score, is used as the evaluation criterion.

$$SR = \begin{cases} s_{own}\frac{(s_{own}-s_{opp})}{(s_{own}+s_{opp})} & \text{if } s_{own} > s_{opp} \\ s_{opp}\frac{(s_{own}-s_{opp})}{(s_{own}+s_{opp})} & \text{if } s_{own} < s_{opp} \end{cases} \quad (1)$$

SR stands for score rate, s_{own} is the own score, and s_{opp} is the opponent score. Results of the 2-minute games played on the empty field, against *NullTeam*, *BrianTeam*, and *MarketTeam* (Kose et al., 2003b; Kose et al., 2003a) are provided in Figure 3. 300 games were played in each configuration, and the average score-rates after every $10th$ game were recorded. The blue dots represent the score-rate whereas the red curves represent the general tendency, which is towards greater score-rates as a sign of scoring more goals and receiving less goals. As seen in the results, there is a general tendency towards scoring more goals (represented as the red curves) while preventing the opponent from scoring.

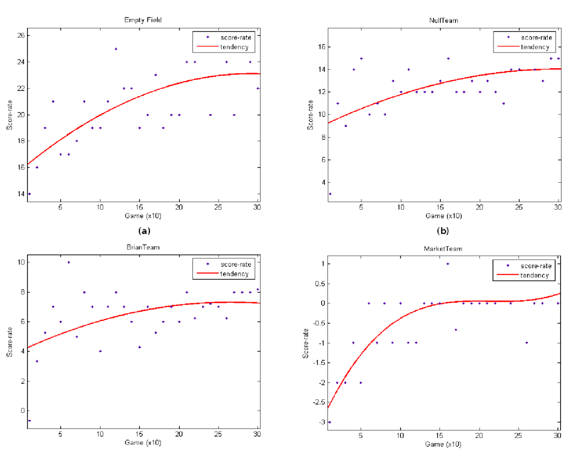

Figure 3. Results of the games played **(a)** on the empty field, then against **(b)** *NullTeam*, **(c)** *BrianTeam*, and **(d)** *MarketTeam*.

4. Conclusions & Future Work

We proposed a biologically inspired approach based on the principles Braitenberg vehicles for creating a team of soccer playing robots that use combinations of very primitive skills to implement "complex" behaviors such as attacking the opponent goal, supporting the attacker, defending the own goal, and passing. We used $Q(\lambda)$ learning to learn the mapping between states, which are represented with only 3 state variables, and one of five possible actions.

The experiments show that our team was able to learn how to score goals efficiently and how to defend their own goal at the end of a short training period, which started on an empty field and ended after playing against a very strong team. The algorithm is implemented in TeamBots simulation environment, in which differential drive robots are used.

This approach can be extended to more than five players by simply defining more targets on the field that the robots can move towards or away from; for instance, the opponent players can be marked in that way. Tests on scalability of this approach are left as future work.

References

Balch, T. (2000). *Teambots mobile robot simulator*. http://www.teambots.org.

Braitenberg, V. (1984). *Vehicles: Experiments in synthetic psychology*. MIT Press / Bradford Books.

FIRA (2010). FIRA - Federation of International Robot-soccer Association. http://www.fira.net/soccer/mirosot/overview.html.

Kaplan, K., & Akin, H. L. (2003). A controller design for soccer-robot teams. *TAINN*.

Kose, H., Kaplan, K., Mericli, C., & Akin, H. L. (2003a). Genetic algorithms based market-driven multi-agent collaboration in the robot-soccer domain. *FIRA Robot World Congress 2003*. Vienna, Austria.

Kose, H., Mericli, C., Kaplan, K., & Akin, H. L. (2003b). All bids for one and one does for all: Market-driven multi-agent collaboration in robot soccer domain. *ISCIS 2003* (pp. 529–536).

RoboCup Small Size League (2010). RoboCup Small Size League. http://small-size.informatik.uni-bremen.de/.

Stone, P., & Veloso, M. (2000). Multiagent systems: A survey from a machine learning perspective (pp. 345–383.).

Sutton, R., & Barto, A. (1998). *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press.