Towards Collaborative and Adversarial Learning: A Case Study in Robotic Soccer

Peter Stone and Manuela Veloso
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
{pstone,veloso}@cs.cmu.edu
http://www.cs.cmu.edu/{~pstone,~mmv}

Abstract

Soccer is a rich domain for the study of multi-agent learning issues. Not only must the players learn to adapt to the behavior of different opponents, but they must learn to work together. We are using a robotic soccer system to study both adversarial and collaborative multi-agent learning issues. Here we briefly describe our experimental framework along with an initial learned behavior. We then discuss some of the issues that are arising as we extend our task to require collaborative and adversarial learning.

Introduction

Soccer is a rich domain for the study of multi-agent learning issues. Teams of players must work together in order to put the ball in the opposing goal while at the same time defending their own. Adaptive learning is essential in this task since the dynamics of the system can change radically as the opponents' behaviors change. The players must be able to adapt to new situations.

Not only must the players learn to adapt to the behavior of different opponents, but they must learn to work together. Soon after beginning to play, young soccer players learn that they cannot do everything on their own: they must work as a team in order to win.

We are using a robotic soccer system to study both adversarial and collaborative multi-agent learning issues. We have had some initial success at using neural networks (NNs) for learning a low-level behavior for a single-agent task in a simulator (Stone & Veloso 1995). We are currently working on improving this behavior while simultaneously extending the task to capture these two types of multi-agent issues.

Here we briefly describe the experimental framework along with the initial learned behavior. We then discuss some of the issues that are arising as we extend our task to require collaborative and adversarial learning.

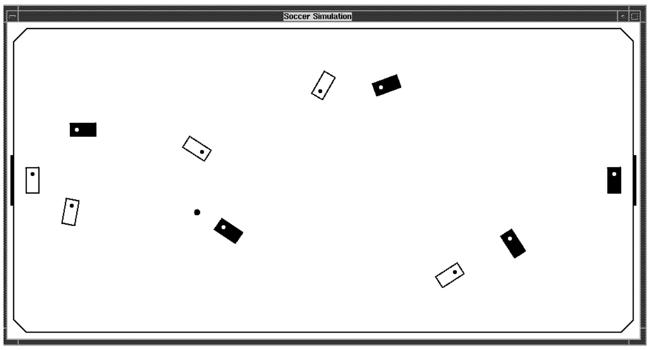
The Simulator

Although we plan to participate in both the real and simulated events of the planned RoboCup event at IJCAI'97 (Kitano *et al.* 1995), the real system of model cars remotely controlled from off-board computers is still under development. Our research is currently conducted in a simulator of the physical system.

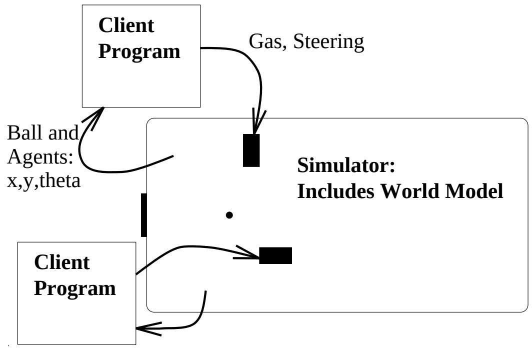
Both the simulator and the real-world system are based closely on systems designed by the Laboratory for Computational Intelligence at the University of British Columbia (Sahota *et al.* 1995). The simulator facilitates the control of any number of cars and a ball within a designated playing area. Care has been taken to ensure that the simulator models real-world responses (friction, conservation of momentum, etc.) as closely as possible. Sensor noise of variable magnitude can be included to model more or less precise real systems. A graphic display allows the researcher to watch the action in progress, or the graphics can be toggled off to speed up the rate of the experiments. Figure 1(a) shows the simulator graphics.

The simulator is based on a client-server model in which the server models the real world and reports the state of the world while the clients control the individual cars (see Figure 1(b)). Since the moving objects in the world (i.e. the cars and the ball) all have a position and orientation, the simulator describes the current state of the world to the clients by reporting the x, y, and θ coordinates of each of the objects indicating their positions and orientations. The clients periodically send throttle and steering commands to the simulator indicating how the cars should move. It is the simulator's job to correctly model the motion of the cars based on these commands as well as the motion of the ball based on its collisions with cars and walls.

For the purposes of this paper, the only path planning needed is the ability to steer along a straight line. This task is not trivial since the client can only control the car's motion at discrete time intervals. Our algorithm controls the car's steering based on its offset from the correct heading as well as its distance from the line to be followed. This algorithm allows the car to be steering in exactly the right direction within a small distance from the line after a short adjustment period. Thus the car is able to reliably strike the ball in a given direction.



(a)



(b)

Figure 1: **(a)** The graphic view of our simulator. Eventually teams of five or more cars will compete in a real-time game of robotic soccer. **(b)** The interface between the clients and the simulator.

Learning to Shoot a Moving Ball

We initiated our research with the task of having an agent learn to shoot a moving ball passed by a passing agent with a simple behavior which could affect the speed and the trajectory of the ball. Hence, in this task there is a single learning agent. We call the learned behavior a *shooting template*. This initial task provided the basis for our interest and on-going research on more elaborate multi-agent learning scenarios. We describe here only enough of the experiments to illustrate the task. For more details, please see (Stone & Veloso 1995).

In all of our initial experiments, a *passer* accelerates as fast as possible towards a stationary ball in order to propel it between a *shooter* and the goal. The resulting speed of the ball is determined by the distance that the passer started from the ball. The shooter's task is to time its acceleration so that it intercepts the ball's path and redirects it into the goal. We constrain the shooter to accelerate at a fixed constant rate (while steering along a fixed line) once it has decided to begin its approach. *Thus the behavior to be learned consists of the decision of when to begin moving: at each action opportunity the shooter either starts or waits.* Once having started, the decision may not be retracted. The shooter must make its decision based solely on the ball's and its own (x, y, θ) coordinates reported at a (simulated) rate of $60Hz$.

Throughout our initial experiments, the shooter's initial position varies randomly within a continuous range: its initial heading varies over 70 degrees and its initial x and y coordinates vary independently over 40 units as shown in Figure 2(a).[1] The two shooters pictured show the extreme possible starting positions, both in terms of heading and location.

Since the ball's momentum is initially across the front of the goal, the shooter must compensate by aiming wide of the goal (by 170 units) when making contact with the ball (see Figure 2(b)). Before beginning its approach, the shooter chooses a point wide of the goal at which to aim. Once deciding to start, it then steers along an imaginary line between this point and the shooter's initial position, continually adjusting its heading until it is moving in the right direction along this line. The shooter's aim is to intercept the ball's path so as to redirect it into the goal.

The task of learning a shooting template has several parameters that can control how hard it is. First, the ball can be moving at the same speed for all training examples or at different speeds. Second, the ball can be coming with the same trajectory or with different trajectories. Third, the goal can always be in the same place during testing as during training, or it can change locations (think of this parameter as the possibility of aiming for different parts of the goal). Fourth, the training and testing can occur all in the same location, or the testing can be moved to a different symmetrical location on the field. Figure 3 illustrates some of these variations. Using a neural network (NN), we have learned a shooting template that can be successful despite variations along all of these dimensions. Table summarizes our initial results.

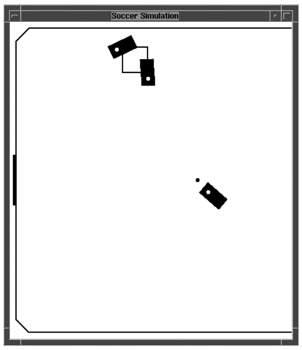
Multi-agent Extensions

For the purposes of this symposium, the task of shooting a moving ball into a goal with a single learning agent in a multi-agent setup is interesting for the opportunities to create multi-agent learning scenarios that it provides. Although players must first learn low-level skills, soccer is inherently a multi-agent task: a team of the best-skilled individual players in the world would be easily beaten if they could not work together. Similarly, an unskilled team that can find and exploit a weakness of a better team, will often be able to prevail. The shooting template can be extended to include both of these higher-level aspects of soccer: cooperative and adversarial.

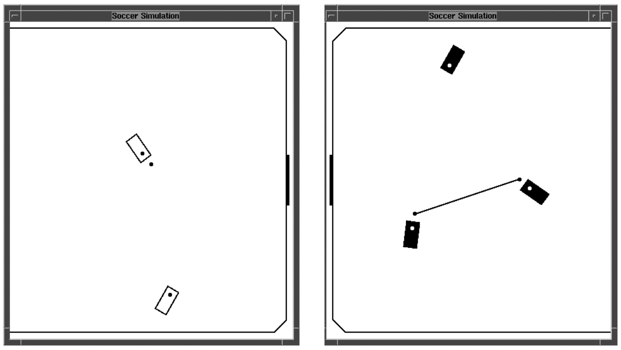
[1] For reference, the width of the field (the side with the goal) shown in figure 2(a) is 480 units. The width of the goal is 80 units.

Shooter Position	Ball Speed	Ball Trajectory	Goal Position	Decision Criterion	Success (%)
Constant	Constant	Constant	Middle	Constant	100
Varying	Constant	Constant	Middle	Constant	60.8
Varying	Constant	Constant	Middle	Random	19.7
Varying	Constant	Constant	Middle	Learned	96.5
Varying	Varying	Constant	Middle	Learned	91.5
Varying	Varying	Varying	Middle	Learned	95.4
Varying	Varying	Varying	Low	Learned	84.1
Varying	Varying	Varying	High	Learned	74.9

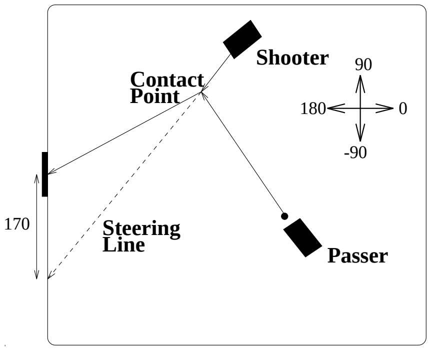
Table 1: A summary of our learning results for shooting a moving ball into a goal. The learned decision policy is trained on the examples generated with the random decision criterion.



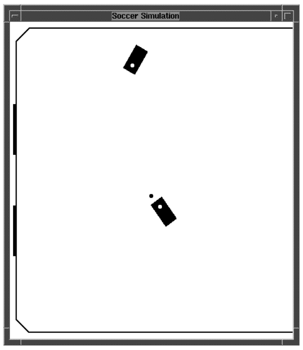
(a)



(b) (c)



(b)



(c)

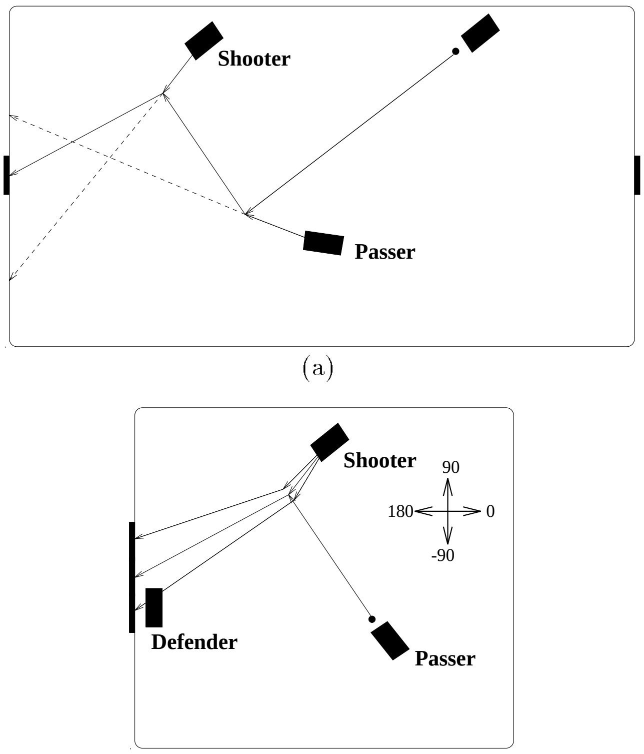
Figure 2: **(a)** The initial position for the experiments. The car in the lower part of the picture, the *passer*, accelerates full speed ahead until it hits the ball. Another car, the *shooter* then attempts to redirect the ball into the goal on the left. The two cars in the top of the figure illustrate the extremes of the range of angles of the shooter's initial position. The square behind these two cars indicates the range of the initial position of the *center* of the shooter. **(b)** A diagram illustrating the paths of the ball and the cars during a typical trial.

Figure 3: Variations to the initial setup: **(a)** The initial position in the opposite corner of the field (a different action quadrant); **(b)** Varied ball trajectory: The line indicates the ball's possible initial positions, while the passer always starts directly behind the ball facing towards a fixed point. The passer's initial distance from the ball controls the speed at which the ball is passed; **(c)** Varied goal position: the placement of the higher and lower goals are both pictured at once.

Cooperative Learning

For the task of shooting a moving ball, the passer's behavior was predetermined: it simply accelerated as fast as it could until it hit the ball. We varied the velocity (both speed and trajectory) of the ball by simply starting the passer and the ball in different positions.

However, in a real game, the passer would rarely have the opportunity to pass a stationary ball that is placed directly in front of it. It too would have to learn to deal with a ball in motion. In particular, the passer would need to learn to pass the ball in such a way that the shooter could have a good chance of putting the ball in the goal (see Figure 4(a)).



(a)

(b)

Figure 4: Multi-agent learning scenarios: **(a)** The passer and the shooter must both learn their tasks in such a way that they can interact successfully; **(b)** The defender learns to block the shot, while the shooter simultaneously tries to learn to score.

Our initial approach to this problem is to take the shooting template as fixed and to try to learn to pass a ball in such a way that this behavior will be successful. However, it may turn out that the passer learns to pass in a non-optimal way. In this case, it will be to the team's benefit if the shooter can learn to shoot the type of moving ball at which the passer is good at providing. Here is a situation in which an adaptive agent will have an advantage over an agent with fixed behavior.

Once the shooter adapts to the passer's behavior, the passer may in turn be able to adjust its behavior to improve their overall performance. Since each player's performance depends on that of the other, we have an interesting example of co-evolution set up here: the shooter and the passer must learn to coordinate behaviors if there is to be any success at all.

Adversarial Learning

On the other hand, the shooting template can be extended to an adversarial situation by putting in a defender. In order to make this task fair, we widen the goal by 3 times (see Figure 4(b)). During our initial experiments, the shooter learned to shoot at the upper and lower portions of the goal as well as the middle. So the shooter can aim away from the defender; the defender must try to figure out where the shooter is aiming and block that portion of the goal.

The defender may be able to learn to judge where the shooter is aiming before the shooter strikes the ball by observing the shooter's approach. Similarly, if the defender starts moving, the shooter may be able to adjust and aim at a different part of the goal. Thus as time goes on the, opponents will need to co-evolve in order to adjust to each other's changing strategies. Note that a sophisticated agent may be able to influence the opponent's behavior by acting consistently in the same way for a period of time, and then drastically changing behaviors so as to fool the opponent.

Multi-agent Learning Methods

Aside from the first scenario in which the shooter's behavior remained fixed, none of the scenarios described in this section can be learned with the supervised learning technique used to learn the shooting template in Section (NNs). Instead, since the agents' behaviors must continually adapt, we need a more incremental type of learning approach. One promising option is to have several trained behaviors with an adaptive decision function wrapped around them to decide which one to use.

For example, there could be a trained template that the shooter learned based on examples from one situation and another based on other examples. Then the shooter's decision function of which template to use in a given situation could be adaptive. In order to make this a truly adaptive method, additional behaviors need to be trained periodically on the newly experienced examples.

Discussion and Conclusion

Robotic Soccer is a rich domain for the study of multiagent learning issues. There are opportunities to study both collaborative and adversarial situations. However, in order to study these situations, the agents must first learn some basic behaviors. Similar to human soccer players, they can first learn to make contact with a moving ball, then learn to aim it, and only then start thinking about trying to beat an opponent and about team-level strategies.

Having achieved some success at learning a low-level behavior we are currently working on extending it to limited multi-agent scenarios. Our future research agenda includes improving the low-level behaviors while simultaneously working on collaborative and adversarial learning issues. We will collect several different behaviors learned in a similar manner to the shooting described in this paper. Then we will use them to learn how to work as a part of a team against hostile opponents. We are currently in the process of implementing our learning techniques in the simulator that will be used in RoboCup '97 (Noda 1995).

References

Kitano, H.; Asada, M.; Kuniyoshi, Y.; Noda, I.; and Osawa, E. 1995. Robocup: The robot world cup initiative. In *IJCAI-95 Workshop on Entertainment and AI/Alife*, 19–24.

Noda, I. 1995. Soccer server : a simulator of robocup. In *Proceedings of AI symposium '95*, 29–34. Japanese Society for Artificial Intelligence.

Sahota, M. K.; Mackworth, A. K.; Barman, R. A.; and Kingdon, S. J. 1995. Real-time control of soccer-playing robots using off-board vision: the dynamite testbed. In *IEEE International Conference on Systems, Man, and Cybernetics*, 3690–3663.

Stone, P., and Veloso, M. 1995. Broad learning from narrow training: A case study in robotic soccer. Technical Report CMU-CS-95-207, Computer Science Department, Carnegie Mellon University. Submitted to the Twelfth International Conference on Machine Learning (ML'96).