Behavior Acquisition and Classification: A Case Study in Robotic Soccer

Patrick Riley and **Manuela Veloso**
Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213

Increasingly in domains with multiple intelligent agents, each agent must be able to identify what the other agents are doing. This is especially important when there are adversarial agents inferring with the accomplishment of goals. Once identified, the agents can then respond to recent strategies and adapt to improve performance.

This research works under the hypothesis that fast and useful adaptation can be done by analogy to previous observations. We introduce methods to extract similarities in temporal observations of the world. First, past observations are organized into a set of behavior classes. By analyzing similarities, the current adversary can be classified into this set of behavior classes. The agents can then employ the most effective strategy against that behavior group.

The test domain for this research is the Soccer Server System (Noda *et al.* 1998) as used in the Robot World Cup Initiative (Kitano *et al.* 1997). The server provides a realistic *simulation* of a soccer game. Distributed software agents interact in a complex, noisy, inaccessible environment. The software was developed based on the champion CMUnited99 agent team (Stone, Riley, & Veloso 2000).

The raw data of the simulation consists of locations of players and the ball over time. The data is first broken into windows of fixed size. For each window, several features are extracted. Each feature extractor watches for a particular type of event (such as an opponent's pass or an opponent's shot). Upon observing an event of the right type, the feature extractor records where on the field, but not when in the window the event occurred.

The recordings of all the games at RoboCup-98 and RoboCup-99 were used as the data sets. A behavior class is created for each team in the competitions. The teams are first observed on a fraction of the games they played. Then, for each type of feature, the data from each window is averaged together to create a "target configuration" for that feature type. In other words, a behavior class consists of a set of examples for what each feature extractor should return if the current opponent is in that class.

After creating these behavior classes, the goal is to correctly identify which teams were playing based on these observations. In order to perform any classification, there must be a notion of similarity between the target configuration and what was actually observed. A novel similarity metric was developed that takes in account spatial localities of topological differences.

Classification was performed in two ways. First with a standard nearest-neighbor approach and then by training a decision tree with the similarities to all of the target configurations as the feature set.

The number of teams in the two competitions was 34 and 37 respectively, making the accuracy of random guessing about 3%. The nearest-neighbor approach performed very poorly on both data sets, with both a 500 cycle window and a 1000 cycle window, doing better than random guessing in only one case. The decision tree approach performed much better (Figure 1). The results also point out an interesting tradeoff in the window length. A long window gives a more accurate sample of the opponent, but a short window gives more data for better learning as well as faster adaptation.

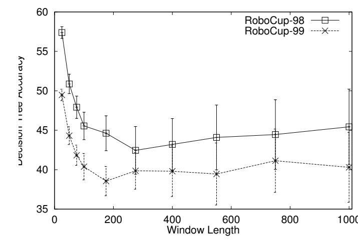

Figure 1: Decision Tree Accuracy

Future work could include smarter creation of behavior classes. Rather than the *a priori* distinction of team name, a clustering approach could be used, such as (Sebastiani, Ramoni, & Cohen 1999). Also, Hidden Markov Models may be useful in capturing more complex events for the features (Han & Veloso 1999). Also, automatically determining the correct window length, perhaps on a per-feature basis, could be very useful.

The main contributions of this research are: a windowing approach to abstracting features in complex domains; a novel discrete spatial similarity metric; and a demonstration that the windowing approach can capture important strategic features in a particular complex, dynamic domain.

References

Han, K., and Veloso, M. 1999. Automated robot behavior recognition applied to robotic soccer. In *Proceedings of IJCAI-99 Workshop on Team Behaviors and Plan Recognition*.

Kitano, H.; Tambe, M.; Stone, P.; Veloso, M.; Coradeschi, S.; Osawa, E.; Matsubara, H.; Noda, I.; and Asada, M. 1997. The robocup synthetic agent challenge. In *Proceedings of IJCAI-95*, 24–49.

Noda, I.; Matsubara, H.; Hiraki, K.; and Frank, I. 1998. Soccer server: A tool for research on multiagent systems. *Applied Artificial Intelligence* 12:233–250.

Sebastiani, P.; Ramoni, M.; and Cohen, P. 1999. Unsupervised classification of sensory inputs in a mobile robot. In *Proceedings of the IJCAI Workshop on Neural, Symbolic, and Reinforcement Methods for Sequence Learning*.

Stone, P.; Riley, P.; and Veloso, M. 2000. The CMUnited-99 champion simulator team. In Veloso; Pagello; and Kitano., eds., *RoboCup-99: Robot Soccer World Cup III*. Berlin: Springer.

Tambe, M., and Rosenbloom, P. 1995. Resc: An approach for dynamic, real-time agent tracking. In *Proceedings of IJCAI-95*.