Active Behavior Learning for Multi-agent Cooperation and Competition

Fahong Li*
Department of Computer Science
University of British Columbia
fhli@cs.ubc.ca

Human active learning typically involves experiences of doing, of observing and of dialog with self and others (Fink 1999). Players in sports games such as soccer and ice hockey cooperate with their teammates and compete with their opponents in a rule-well-defined world but various outcome-quite-uncertain game situations. They learn play skills, tactics and strategies by practice, mainly from coaches, peers and guidance materials such as books, videotapes, etc. While learning, they usually follow some particular drills to achieve specific objectives; if necessary they also design (and practice) new drills to achieve the same or new objectives. Sports game analysis summarizes effective play experiences into tactics or strategies, and reiterates the process when the tactics or strategies are applied back to the game play. It also examines the learning process of specific players to particular tactics and provides individual training guidance. This sports game analysis domain provides rich enough controllable and tractable contexts for testing ideas in building intelligent systems.

Machine learning techniques are widely researched and many have been applied to real-world problems. One influential factor in their success or failure is the availability and the suitability of training data. In some cases, the training data set is given or fixed, and how to use these data is part of the particular learning technique. For instance, given the freedom of choosing future training data based on those that have been used, (Tong 2001) proposes a general approach for active learning to minimize the number of training data required and thus to reduce the cost of gathering them. This approach first defines a model and an associated model quality or loss function appropriate for the learning task at hand. Then it chooses a method to compute the potential model loss given a potential query, and asks the query which causes the lowest potential model loss. It has been applied in three areas of machine learning: classification with Support Vector Machines, parameter estimation and causal structure discovery in static Bayesian Networks. Empirical results show the active learning technique can significantly reduce the need for training data. There are still many aspects for the active learning method to explore, such as dealing with situations involving missing data values, hidden variables, high-dimensional problems and temporal domains.

In many other cases, gathering training data itself is an essential part of the learning task. For example, we argue that in the domain of RoboCup Soccer 2D Simulation (RCS2DS) (Chen *et al.* 2003) agents representing individual players need to design their own drills to collect the training data for learning various low-level soccer skills and high-level tactics or strategies. We define this kind of learning process, i.e., applying existed guidance plans (including skills, tactics and strategies) and innovating new ones appropriately to achieve specific learning objectives, as "Active Behavior Learning (ABL)." ABL includes the design of automatic means to gather training data according to the purposes of the learning task. In the domain of RCS2DS, it involves automatically building dynamic and effective mappings between agents' high-level abstract/compact play strategies and their low-level concrete/primitive action sequences. The mapping process is two-direction, i.e., from strategies to action sequences (applying/planning) and the reverse (summarizing). Trial and error will be used in the process. One of the difficulties lies in designing good trials to generate concrete, reasonable and efficient action sequences for multiple agents, which cooperate through limited communication while playing against adaptive opponents. Automatically collecting/correcting errors and summarizing action sequences into strategies is also challenging.

While ABL is trying to make agents learn as human does in some aspects, many other approaches have been tested in RCS2DS. (Stone 2000) presents a flexible team member agent architecture and a general-purpose layered machine learning paradigm which allows learning at each level of a hierarchy composed of complex tasks. Roles, formations and set-plays are used to compose teamwork structure, and paradigms of communication among agents with a single unreliable low-bandwidth channel are also analyzed. In the layered learning paradigm: neural network is used to learn an individual soccer skill, ball interception; decision tree is employed to learn a multi-agent behavior, pass evaluation; and a new algorithm TPOT-RL (Team-Partitioned, Opaque-Transition Reinforcement Learning) is taken to learn a team behavior, pass selection. (Reis, Lau, & Oliveira 2001) presents a formalization of team strategy and concepts of Situation Based Strategic Positioning (SBSP) and Dynamic

*Supervised by Dr. Robert J. Woodham. Phone #: 604-827-3986. URL of complementary material: http://www.cs.ubc.ca/∼fhli/abl.html. Postal address: Rm 201 2366 Main Mall Vancouver BC, Canada V6T 1Z4

Positioning and Role Exchange (DPRE) for homogeneous agents to collaborate against opponents in dynamic, real-time and uncertain environments. The formalized team strategy is composed of a set of tactics and several possible agent types. A tactic consists of formations that are applied in different situations. A formation assigns each agent its agent types and its positions in the field. The strategic positioning of one agent depends on the situation and the positioning of other agents assigned in the formation. Agents take more reactive behaviors, i.e., domain-specific high-level and low-level skills, when they are in active situations, which are identified out of strategic situations. Agents are also able to switch their positions and roles (specific behaviors) at runtime, within one formation. A high-level decision module is used to decide an agent's current tactic, formation, role and action at a given moment. The FC Portugal RoboCup Soccer team implemented an agent architecture embodying this team strategy and concepts of SBSP and DPRE. It won the World RoboCup 2000 without losing any goal.

(Montgomery 2003) introduces a constraint-based agent architecture to address the issues of an agent's synchronizing with, reacting to, and operating proactively and appropriately in a dynamic environment. Intentions represented as constriants are attributed to the agent to achieve its resource-bounded deliberation. The architecture adaptively schedules deliberation processes so that the agent can evolve its action with the natural frequency of the environment's dynamics and produce quality-varying solutions. A theory of intentionality abstracts behaviors and facilitates agents' behavior generation and recognition. The architecture is implemented and tested in RCS2DS but the internal representation model has yet been integrated into the whole decision process of the agents. (Riley 2005) defines the coaching problem as an automated coach agent providing advice to one or more automated advice-receiving agents in RCS2DS, and explores its solutions, such as learning and using models of the environment, adapting advice to receivers' peculiarities, representing advices and modeling opponents. The author co-developed the standard advice language "CLang" and proposed a multi-agent plan representation "MASTN (Multi-Agent Simple Temporal Network)", together with an associated distributed plan execution algorithm. However, in the current implementation of the algorithm, the agents do not take over actions which were originally assigned to others but not executed successfully.

We propose a proof of concept system for cooperative, multi-agent active behavior learning in adversarial environments using the RCS2DS environment as our test-bed. It has several components as shown in Figure 1. We provide agents with prior domain knowledge about the environment and built-in mechanisms to maintain (create, update and remove) that knowledge. According to the prior domain knowledge and the current learning task (e.g., a shooting practice of 2 attackers against 1 defender), each agent identifies its own objectives (including those shared with teammates), designs its own or shared plans, executes them while obtaining feedback from the environment. Feedback includes that given explicitly by other agents, if any. The agent then updates the action plans according to the feedback and

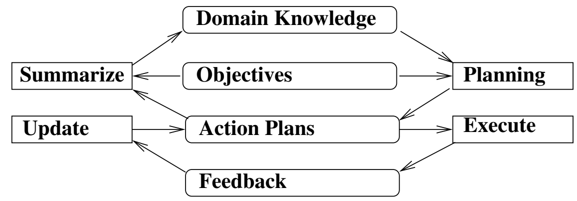

Figure 1: Framework components

executes them again, until it achieves all the learning objectives. Finally it summarizes the action plans and the learning task (or the learning objectives) into knowledge items and merges them into its domain knowledge. We believe that active behavior learning can significantly reduce the learning time while maintaining (or exceeding) a given level of task performance. A prototype of our proposed framework for sports game analysis will be able to demonstrate various play strategies of autonomous agents under different situations. It severs the task of coaching players at different levels of play skills. The technologies supporting the agents' planning, learning and reasoning can be generalized and applied into robotics and computer games.

References

Chen, M.; Dorer, K.; Foroughi, E.; Heintz, F.; Huang, Z.; Kapetanakis, S.; Kostiadis, K.; Kummeneje, J.; Murray, J.; Noda, I.; Obst, O.; Riley, P.; Steffens, T.; Wang, Y.; and Yin, X. 2003. *Users Manual RoboCup Soccer Server for Soccer Server Version 7.07 and later*. The RoboCup Federation. Retrieved from `http://umn.dl.sourceforge.net/sourceforge/sserver/manual-20030211.pdf` on Jan 24, 2007.

Fink, L. D. 1999. Active learning. Retrieved from `http://honolulu.hawaii.edu/intranet/committees/FacDevCom/guidebk/teachtip/active.htm` on Jan 24, 2007.

Montgomery, J. 2003. Situated observation and participation in multiple-agent systems. Master's thesis, Computer Science Dept., The University of British Columbia.

Reis, L. P.; Lau, N.; and Oliveira, E. C. 2001. Situation based strategic positioning for coordinating a team of homogeneous agents. In Hannebauer, M.; Wendler, J.; and Pagello, E., eds., *Balancing Reactivity and Social Deliberation in Multi-Agent System – From RoboCup to Real-World Applications*, number 2103 in Springer's Lecture Notes in Artificial Intelligence. Berlin: Springer. 175–197.

Riley, P. 2005. *Coaching: Learning and Using Environment and Agent Models for Advice*. Ph.D. Dissertation, Computer Science Dept., Carnegie Mellon University. CMU-CS-05-100.

Stone, P. 2000. *Layered Learning in Multiagent Systems: A Winning Approach to Robotic Soccer*. MIT Press.

Tong, S. 2001. *Active Learning: Theory and Applications*. Ph.D. Dissertation, Stanford University.