Multiagent Learning Systems and Expert Agents

Claudia V. Goldman* Jeffrey S. Rosenschein
Institute of Computer Science
The Hebrew University
Givat Ram, Jerusalem, Israel
clag@cs.huji.ac.il, jeff@cs.huji.ac.il

Abstract

This paper focuses on two main research topics we are investigating. First, we investigate how agents can learn strategic behavior in a teacher-learner model. The notion of the teacher here should be understood as a "trainer". We present the general teacher-learner model together with results from experiments performed in the traffic lights domain.

Second, we investigate how agents can learn to become experts, and eventually organize themselves appropriately for a range of tasks. The model is based on evolutionary processes that lead to organizations of experts. In our case, the organization emerges as a step prior to the execution of a task, and as a general process related to a range of problems in a domain. To explore these ideas, we designed and implemented a testbed based on the idea of the game of *Life*.

Introduction

The focus of my research is in understanding and in experimenting with intelligent agents that learn how to behave, and agents that learn to become experts.

In the first case, the agent learns directly from the other agents it interacts with, in the environment in which it acts. We first studied (Goldman & Rosenschein 1996b) how agents could learn from other agents by receiving training examples, and then generalizing the knowledge they have acquired. Then, we defined the teacher-learner model (Goldman & Rosenschein 1996a), in which each agent plays both roles, the teacher and the learner. The agents learn to adapt to each other dynamically while they actually interact and perform their actions. We consider different types of agents, such as cooperative, altruistic or selfish agents, and then investigate how this influences the adaptation process. Agents that execute actions based on the teacher-learner model shape their behavior according to the other agents' reactions to their current behavior.

*Supported by the Israeli Ministry of Science

We are also interested in agents that "live" in a multiagent system, and learn to become experts. The social expertise of the agents is shaped by their environment. We have developed a model and an algorithm for evolving a population of agents in a given domain (Goldman & Rosenschein 1997;). The implementation of this testbed is along the lines of the Game of *Life* (Gardner 1983). We consider finite and structured domains (e.g., a collection of documents (i.e., html files, mail files,...), or a collection of pre-computed plans for achieving goals in a domain. The aim of the algorithm is to divide the basic units of expertise in the given domain among a number of agents, in such a way that this number is balanced.

We have also investigated a system of four autonomous software agents, that together assist one of them in learning about a concept in the context of a collection of documents (i.e., the agent learns the common knowledge existing in a group of experts in a specific domain). We have implemented this system, called Musag, on the Internet (Goldman, Langer, & Rosenschein 1997). A discussion of this system is beyond the scope of this paper.

Learning and Multiagent Systems

Two different research directions can be distinguished when combining learning and multiagent systems: *learning in a multiagent system* (the agent gets feedback from the world that might include other agents (Mataric 1994b; Sen, Sekaran, & Hale 1994; Asada, Uchibe, & Hosoda 1995)), and *multiagent learning systems* (the agents learn directly from each other (Littman 1994; Mataric 1994a)).

We are interested in autonomous, adaptive, and heterogenous agents that learn to coordinate their courses of actions, and learn to cooperate to avoid conflicts. We propose a trainer-learner model where agents play both roles and learn to coordinate their behaviors when interacting in the same environment. The trainer tries to train the learner to behave for its (i.e., the trainer's) own benefit, while the learner behaves as a *provoking agent*. Each agent learns to adapt to the other agent by evaluating the feedback values given by its teacher, and by computing a function that represents the agent's level of satisfaction. This function depends on the learner's gain from the action it has performed, and the degree of approval of the trainer according to the learner's behavior so far. Each term is weighted based on the learner's type. The learner keeps choosing the same action as long as its satisfaction level does not fall below a certain threshold. This threshold is the limit under which the agent compromises.

An interesting domain in which we can investigate multiagent coordination strategies is the traffic domain. Having agents controlling an intersection of roads automatically and adaptively will lead to a better flow of cars in the roads, given the constraints of the road in real time. We start our experiments with a single traffic intersection: A_1 is responsible for the traffic flow in the vertical direction, while A_2 is in charge of the horizontal road. At time t, each agent knows its state $x^t_{A_i}$ (i.e., how many cars A_i has at time t), and what is the reaction of its teacher to the learner's behavior so far.

We first experimented with two agents, one holding the green light and the other the red light. The agents adapt to each other by learning when to pass the green light to the other agent. When the learner gives up the green light, it becomes the teacher and the other agent the learner. We have tested three types of learning agents: selfish, cooperative, and altruistic.

When both agents are cooperative, they manage to maintain the smallest number of cars in both roads, compared to the performance of random agents and controlled agents. When one of the two agents is selfish, and the other is of another type, then our model has been worse than all the other cases. The relation between all the other runs points out that when both agents are cooperative they coordinate better than when both agents are altruistic or one is cooperative and the other is altruistic. When both agents behave selfishly, they coordinate better than cases where only one agent is selfish. Regarding each road separately, then only when both agents were cooperative were both graphs more balanced, and similar to one another. If one of the agents was selfish, then its road was almost empty. If both agents were selfish, only one road was close to empty and their graphs complement each other. When one of the agents was cooperative, and the other was altruistic, the road of the cooperative agent held less cars. When both agents were altruistic, we got in six cases that A_1's road was close to 0, and in the other six cases A_2's road was more empty. When one agent was selfish, and the other cooperative, then the selfish road held less cars.

For an agent to learn about its type, we have first set the type of agent A_2 to be constant throughout the simulation, and let A_1 learn what should be its type. When the winner type was chosen according to the type that gave the agent the largest time with the green light, the results showed that A_1 should behave selfishly no matter what A_2's type. When the winner's type was picked to be the one that lead the agent to have the green light for the time closer to $\frac{MAXRUN}{2}$, A_1 should behave exactly with the same type as A_2. When A_1 learns its type dynamically during a run, and A_2 is set to behave altruistically or cooperatively, A_1 learned to behave selfishly for intervals equal to ten to fifty ticks of time. When A_2 was selfish, A_1 learned to behave altruistically. When both agents learn each one's type, A_1 ended up being selfish, and A_2, altruistic in nine cases, and in the other three cases, A_1 was cooperative, and A_2 was also altruistic.

In the second scenario, both agents play the role of the learner and the teacher at the same time. For these experiments, another term was added to the satisfaction level function, to punish the agents when they collide or block the roads. When one agent is altruistic and the other is selfish, we get perfect coordination. In all the other cases, there are points of miscoordination. Both agents that cooperate coordinate better than A_1 behaving cooperative and A_2 selfish. These achieve better coordination than A_1 cooperating and A_2 being altruistic. This was better than A_1 choosing to be selfish and A_2 cooperating, or both agents behaving selfishly. All the other cases consisted of more points of miscoordination.

The Organization of Agents

We are interested in the process of evolving an organization of agents as a prior step to the achievement of a goal. This emergent organization can serve as a source of information, and as a multiagent system that can itself solve problems. As a source of information, the organization might be approached by a user or other agent applications looking for information in a given domain (e.g., as an information retrieval system, as a library of reusable plans, as a collection of software tools). Moreover, the organization itself can be given a problem to solve in the domain the agents have information about. Each agent can suggest an initial solution based on its knowledge.

We have designed and implemented a testbed to experiment with evolving organizations of agents. The rules of our testbed were designed along the lines of the game of *Life*. The main addition to the game is the consideration of the environment in which the agents grow, die, or live. In our current implementation we choose the information domain, consisting of documents from a given source.The rules of death, birth, and survival were defined based on the resources that the agents hold. We deal with homogeneous agents coming from a general class of agents. There are two main concepts that need to be defined in the game of *Life*: the neighborhood relation and the rules of the game. We compute the nearness among the documents in the set on which the simulation is run. The evolutionary engine program simulates a system populated with agents that gather resources. The main algorithm evolves a dynamic population of agents that can reproduce, die, or just add documents to their collection after they compared the number of documents they are holding and the current population density of agents on documents that are neighbors of the documents in their collection (more details about this algorithm can be found in (Goldman & Rosenschein 1997)).

There were three known behaviors that could emerge in different configurations with these rules: stable populations, that did not change their structure once they got to it, configurations that faded away, and periodic or oscillating configurations. These examples stress the unpredictability of the results in *Life* even though the rules of the game are very simple, and seem predictable. We found similar results in our testbed, including also the domain we choose.

Conclusions

Our research deals with agents that learn to coordinate their actions by teaching each other what is their reaction to the others' actions, and by learning how to behave based on their level of satisfaction. The emphasis in this work is in the direct learning between the agents, opening the possibility of considering different types of agents that can have different influences as teachers and as learners.

We also deal with agents that learn from the world or domain of action. A population of agents evolves dynamically for a given domain, and divides the basic pieces of expertise in the domain among the agents. Eventually, this organization of agents could serve as a source of information, or as a preparatory step for a division of labor.

References

Asada, M.; Uchibe, E.; and Hosoda, K. 1995. Agents that learn from other competitive agents. In *Machine Learning: Proceedings of the Twelfth international conference, Workshop on Agents that Learn from Other Agents.*

Gardner, M. 1983. *Wheels, Life and other mathematical amusements.* W.H. Freeman and Company.

Goldman, C. V., and Rosenschein, J. S. Patterns of organization in multiagent systems. *Submitted to the International Journal of Applied Intelligence, a special issue on Intelligent Adaptive Agents.*

Goldman, C. V., and Rosenschein, J. S. 1996a. Incremental and mutual adaptation in multiagent systems. Technical Report 96-15, The Hebrew University.

Goldman, C. V., and Rosenschein, J. S. 1996b. Mutually supervised learning in multiagent systems. *Adaptation and Learning in Multi-Agent Systems* LNAI 1042:85-96.

Goldman, C. V., and Rosenschein, J. S. 1997. Evolving organizations of agents. Technical Report WS-97-03, American Association for Artificial Intelligence. Multiagent Learning Workshop at AAAI97.

Goldman, C. V.; Langer, A.; and Rosenschein, J. S. 1997. Musag: an agent that learns what you mean. *Journal of Applied AI, a special issue on Practical Applications of Intelligent Agents and Multiagent Technology* 11(5,6).

Littman, M. L. 1994. Markov games as a framework for multi-agent reinforcement learning. In *Machine Learning: Proceedings of the Eleventh international conference*, 157—163.

Mataric, M. J. 1994a. Learning to behave socially. In *Proceedings From Animals to Animats 3, Third International Conference on Simulation of Adaptive Behavior*, 157—163.

Mataric, M. J. 1994b. Reward functions for accelerated learning. In *Machine Learning: Proceedings of the Eleventh international conference*, 157—163.

Sen, S.; Sekaran, M.; and Hale, J. 1994. Learning to coordinate without sharing information. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, 426-431.