USING MACHINE LEARNING TO EXTEND AUTONOMOUS AGENT CAPABILITIES

W. Lewis Johnson and Milind Tambe
USC / Information Sciences Institute & Computer Science Dept.
4676 Admiralty Way, Marina del Rey, CA 90292-6695
WWW: http://www.isi.edu/soar/{johnson, tambe}
{johnson,tambe}@isi.edu

To appear in the Procedings of the 1995 Summer Computer Simulation Conference.

Keywords: machine learning, interactive simulation, military

1 Introduction

The Soar/IFOR project is developing human-like, intelligent agents that can interact with humans, and with each other, in battlefield simulations [10]. Our agents play a variety of roles such as fighter pilots, helicopter pilots, and airspace controllers. The fighter pilot agents in particular have been successfully deployed in large-scale simulation exercises, such as the Synthetic Theater of War (STOW) exercise in November, 1994, which modeled a four day battle scenario involving approximately 2000 military vehicles. Autonomous agents such as Soar/IFOR agents are expected to continue to play a major role in battlefield simulations, which in turn are expected to provide an essential tool for military planning and training in the future.

Soar/IFOR agents are implemented in Soar, a problem solving architecture that integrates a number of human cognitive functions, including problem solving, perception, and learning [4]. Learning occurs through the application of a general mechanism called *chunking* that summarizes the results of processing on subgoals, in the form of rules that can apply to similar subgoals in the future. This chunking process is a form of explanation-based learning EBL [7, 6]. Chunking can lead to speedup in learner performance, and is instrumental to the learning of new concepts. Some Soar systems have managed to learn thousands, and even hundreds of thousands, of chunks[2].

From the previous experience with learning in Soar, it was taken as a given that the Soar/IFOR agents could be made capable of applying chunking in service of their performance requirements. The first research question that we focus on in this paper is then the following: What kinds of knowledge can Soar/IFOR agents learn in the combat simulation environment? In our investigations so far, we have found a number of learning opportunities in our systems, which yield several types of learned rules. For example, some rules speed up the agents' decision making, while other rules reorganize the agent's tactical knowledge for the purpose of on-line explanation generation.

Yet, it is also important to ask a second question: Can machine learning make a significant difference in Soar/IFOR agent performance? The main issue here is that battlefield simulations are a real-world application of AI technology. The threshold which machine learning must surpass in order to be useful in this environment is therefore quite high. It is not sufficient to show that machine learning is applicable "in principle" via small-scale demonstrations; we must also demonstrate that learning provides significant benefits that outweigh any hidden costs.

Thus, the overall objective of this work is to determine how machine learning can provide practical benefits to real-world applications of artificial intelligence. Our results so far have identified instances where machine learning succeeds in meeting these various requirements, and therefore can be an important resource in agent development. We have conducted extensive learning experiments in the laboratory, and have conducted demonstrations employing agents that learn; to date, however, learning has not yet been employed in large-scale exercises. The role of machine learning in Soar/IFOR is expected to broaden as practical impediments to learning are resolved, and the capabilities that agents are expected to exhibit are broadened.

2 The Problem Domain

Soar/IFOR agents are designed to work within distributed interactive simulations (DIS) of military exercises. But unlike conventional "semi-automated" entities in distributed simulations, Soar/IFOR agents are fully capable of autonomous decision making without outside human intervention. They are intended to be realistic models of military agent behavior, so much so that to an outside observer their behavior is indistinguishable from that of people. They must perform most if not all of the functions that human personnel would be called upon to perform, e.g., to issue and/or understand commands, to coordinate their activities with friendly forces, and to interpret and respond to the actions of enemy units. Needless to say, achieving these goals successfully is a significant achievement for artificial intelligence.

Soar/IFOR agents interact with distributed simulations via the ModSAF simulation package [1]. Each agent is assigned to a ModSAF simulation of a vehicle, e.g., an aircraft. Soar/IFOR receives inputs from the vehicle, via an abstract interface [8], information similar to what a human controlling the same vehicle in the real world would receive, such as position of the vehicle, presence of enemy vehicles in the area, etc. The Soar/IFOR agent interprets the situation based upon the information received, decides on actions to take, and communicates these to ModSAF as commands for the vehicle to execute. Some of the details of psychomotor control and resource contention are omitted from the model, e.g., a Soar/IFOR pilot controls its aircraft by specifying desired altitudes and headings instead of by simulating stick movements. However, these abstractions do not simplify the agents' decision making task.

Soar/IFOR has been tested in simulated exercises incorporating manned simulation devices such as flight simulators, semi-automated forces, as well as automated forces. Soar/IFOR agents are assigned missions prior to the engagement, and are otherwise left to carry out their missions themselves. Agents are evaluated according to how appropriately they perform in each individual engagement.

Although such exercises are useful for demonstrating agent capabilities, they do not in themselves ensure that Soar/IFOR agents meet the needs of potential users of distributed simulations. For example, in order for users to be certain that agent decision making is realistic, they need to understand the rationales for the agent's decisions. This has led to the development of an automated explanation capability, called Debrief, that enables users to engage agents in a question-answer dialog, in a manner analogous to an after-action review [3].

3 Learning in Soar Agents

The air-combat simulation environment—by virtue of its complex, real-world characteristics—presents Soar/IFOR agents with a number of challenging functional and performance requirements. There are also many ways in which machine learning can help the agents meet these requirements. Chunking in IFOR has been found so far to enable the following functional capabilities and performance improvements.

- Decision making speeds up over time.
- A memory of past episodes is maintained.
- Problem solving knowledge is reorganized in order to support explanation and efficient execution.
- Interpretation of situations and events improves in quality with experience.

A Soar/IFOR agent engages in some of this learning *on-line*, i.e., while it is engaged in simulated combat. Prime candidates for such on-line learning include chunking for speedup, episodic memory and knowledge compilation. However, not all learning can or should occur on line. In particular, some of the learning requires that a Soar/IFOR agent consider the consequences of its decisions, explore alternative decisions, and learn from the results. Because of the real-time pressures of air-to-air combat, a Soar/IFOR agent may not have the free time to engage in such deliberation. Time pressures are certainly not continuous: there can be momentary lulls in activity that could be used for deliberation and learning, but as yet are not. Instead, Soar/IFOR agents rely upon *off-line* analysis for such learning. It waits for the combat situation to terminate, so it can analyze past situations without interruption. This enables the agents to explain their reasoning during after-action review, for example.

Learned chunks are applied to future decisions in the following ways. A chunk learned during an engagement may apply later on within the same engagement. It may apply during after-action review of the engagement. Finally, chunks created during a mission or during after-action review are saved so that they can be employed by agents in future missions and review sessions, enabling the agents to learn from accumulated experience.

3.1 Speeding Up Decisions

In much machine learning research, such as [5], speedup is measured by comparing problem solving time after learning to problem solving time without learning. Such a measure is inappropriate for learning in Soar/IFOR, because chunking does not yield an overall speedup, i.e., it does not reduce the overall duration of the engagement. In other domains such lack of speedup might be attributable to the high cost of matching and retrieving the learned chunks[11]. However, for Soar/IFOR agents, the cost of matching and retrieving learned rules is not much of an overhead. Rather a combination of the following two effects are at work. First, combat simulation involves performing (simulated) physical actions and responding to external events. Learning cannot affect the duration of such actions and events; at best it can reduce the time required to decide on an action or interpret an event. Second, cognitive activity is concentrated in isolated episodes, separated by periods of relative inactivity. Speedups in deliberation contribute very little to reductions in the overall duration of a scenario. For instance, suppose a Soar/IFOR agent decides to launch a missile at an opponent. To that end, it must decide which type of missile to employ, and how best to approach the opponent's aircraft. These decisions take up at most a few seconds. The agent then has to wait, sometimes for up to a minute or more while the opponent gets into its missile firing range. Decision time thus has little or no effect on overall time to intercept the opponent.

Although learning has little effect on the overall duration of engagements, it can make a substantial difference in time-critical situations. In such situations, small delays in an agent's action can jeopardize its survival, or prevent the agent from exploiting momentary advantages over an opponent. For instance, when a Soar/IFOR agent fires a missile at its opponent, the opponent may engage in a missile evasion tactic that can cause it to break radar contact (disappear from the Soar/IFOR agent's radar). The opponent may then turn quickly to fire a missile at the Soar/IFOR agent. This is an extremely time-critical situation. When the opponent turns back after its missile evasion maneuver, the Soar/IFOR agent obtains a new contact (blip) on its radar. This blip could be the opponent, or perhaps a friendly aircraft who has just arrived in radar range. The Soar/IFOR agent must quickly determine the contact's identity, and then launch a second missile before the opponent fires her missile. If the Soar/IFOR agent is delayed in re-establishing the opponent's identity, it may get shot down. Chunking can enable Soar/IFOR agents to arrive at important decisions more rapidly the next time a similar situation is encountered. The end result is that the agents can survive longer, and fight better.

A possible way of measuring speedup might be to measure an agent's reaction time, i.e., the from an external event until the agent's response to that event. This presupposes, however, that the stimuli are controlled so that there is a clear relationship between stimulus and response. However, battlefield engagements are not like controlled laboratory experiments: instead, agents are constantly exposed to a variety of stimuli, and perform a variety of tasks, often at the same time. Reducing the amount of time required to interpret one stimulus often has the indirect effect of enabling the agent to attend to other stimuli that were previously overlooked, such as a second opponent that has just arrived in radar range. This clearly can have an impact on overall agent performance, but in a way that is difficult to quantify.

3.2 Maintaining an Episodic Memory

It is useful for Soar/IFOR agents to have an episodic memory, so that they can recall episodes from previous engagements during after-action review or subsequent missions. Episodic memory can be regarded as an aspect of learning, insofar as the problem solver's reasoning after memory formation is different from that before memory formation. It is instrumental to other types of learning: for example, if an agent can recognize that the current situation is similar to previous situations, it can then apply its previous experience to the new situation.

We have found that chunking can be readily employed to address part of the episodic memory problem, namely to learn to recall the circumstances in which a given event occurred. That is, when presented with a description of an event, chunks fire which recreate a description of the world state that prevailed at that time. Other aspects of episodic memory, such as recalling what events occurred as part of a given mission, are not as yet handled via chunking; the agent instead simply records the events that occur in a conventional list data structure.

The episodic memory mechanism relies on two sets of chunks. The first set consists of *recognition chunks*, which are common in a range of Soar systems. Recognition chunks fire in response to some description that serves as a memory probe, indicating that an instance

matching the probe has been seen before. In the Soar/IFOR case, the memory probe consists of a description of an event, together with a possible state change. If the state change occurred at the time the event was observed, the recognition chunk will fire. These recognition chunks are created in a special episodic-memory subgoal, which is processed whenever the agent notices a significant state change. The second set of chunks are *recall chunks*, which recall the complete state in which an event occurred, when presented with an event description as a memory probe. The first time Soar/IFOR attempts to recall the state associated with an event, it first tries to find an earlier event for which it can recall a state. It then tries to recall which state changes occurred between the earlier state and the state of interest. The previously created recognition chunks identify the relevant state changes. Once the recall process is complete, a recall chunk is created, so that the next time the event is used as a memory probe the state is immediately recalled.

Episodic memory illustrates how chunking can serve as an underlying mechanism for a variety of types of learning besides simple speedup. Such learning may require problem spaces that are specially designed to generate particular types of chunks such as recognition chunks or recall chunks.

3.3 Reorganizing Knowledge

Chunking also enables Soar/IFOR agents to reorganize their knowledge. In knowledge based systems generally, the form in which knowledge is encoded depends upon how the knowledge engineer intends the knowledge to be used. Learning enables knowledge encoded for one purpose, i.e., controlling the agent's behavior, to be employed for other purposes, e.g., explaining the agent's decisions.

Soar/IFOR's interactive explanation capability, called Debrief, makes extensive use of chunking for knowledge reorganization [3]. The agents can explain the rationales for decisions made during an engagement, by relating chosen decisions to the critical factors in the situation that led to those decisions. The knowledge needed to generate such explanations, i.e., associations between decisions and sets of situational factors, is different from the knowledge used to generate the decisions in the first place. For one thing, the process of generating the decision may involve internal reasoning mechanisms that are of little interest to someone who is not an agent developer. Recognition chunks are built which identify the key factors leading to a decision in a given situation. This is accomplished by reconsidering the decisions after the engagement is over, and proposing hypothetical changes to the situation in which the decision was made. The set of state features that prove significant, because altering them alters the outcome of the decision, is saved in a chunk. If the agent is asked to explain a similar decision in a similar situation, the recognition chunk will fire identifying those features of the situation that should be included in the explanation.

Knowledge reorganization also allows knowledge organized for ease of knowledge engineering to be rendered in a form suitable for efficient execution. The Soar/IFOR project is developing a variety of types of agents, among which only some knowledge is shared. Rules therefore tend to be factored so as to separate the shared knowledge from the unshared knowledge. Chunking is used in some cases to combine this knowledge into larger agent-specific rules, thus reducing the number of rules that must execute. This happens because chunking summarizes the results of all rules that are executed in a subgoal, in the form of a single rule that represents their combined effect. Agent developers are thus free to encode the knowledge in a factored form, with the expectation that the factored rules will be combined when they are executed by the agent.

3.4 Improving Situation Interpretations

Accurate interpretations of the rapidly evolving battlefield situation is key to a Soar/IFOR agent's successful task performance. One important component of such an interpretation is accurate tracking of an opponent's ongoing actions, to infer her higher level goals, plans or behaviors. For instance, a Soar/IFOR agent cannot actually observe a missile, but needs to infer a missile firing based on the opponent's maneuvers, as shown in Figure 1. Here, the Soar/IFOR agent is piloting the dark-shaded aircraft and its opponent the light-shaded one. In Figure 1-a the two aircraft are on collision course—if they fly straight they will collide at the point shown by x. After reaching her missile firing range, the opponent turns her aircraft to point at the Soar/IFOR agent's aircraft (see Figure 1-b). In this situation, the opponent fires a missile. She then turns 45-degrees—an *Fpole* turn—to provide radar guidance to the missile, while slowing the closure between the two aircraft. The Soar/IFOR agent cannot observe this missile, but based on the opponent's turn to point at its aircraft and the subsequent Fpole turn, it needs to infer that the opponent has fired a missile.

Unfortunately for the Soar/IFOR agents, the hu-

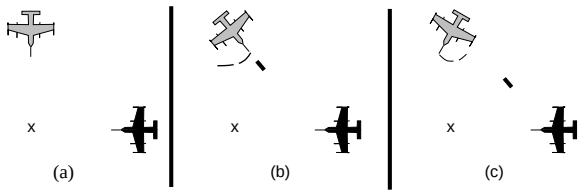

Figure 1: Tracking an opponent's normal missile firing maneuvers. An arc on an aircraft's nose indicates its turn direction. The missile is indicated by –.

man pilots in the STOW-E exercise (see Section 1) were briefed as to what cues Soar/IFOR looks for when interpreting opponent actions, and how they might be able to fool Soar/IFOR by avoiding these cues. They deliberately modified their missile firing behavior to fire missiles while maintaining a 25-degree angle-off (i.e., pointing 25-degrees away from Soar/IFOR agents' aircraft). The Soar/IFOR agents failed to track the missile firing and got shot down. Of course, human pilots are bound to come up with novel variations on known maneuvers, and the Soar/IFOR agents cannot be expected to anticipate them. Yet, at the same time, agents cannot remain in a state of permanent vulnerability—for instance, getting shot down each time the variation of 25-degrees gets used—otherwise they would be unable to provide a challenging and appropriate training environment for human pilots.

The Soar/IFOR agents must adapt their opponent tracking to counter such adaptive behavior on the part of humans. To this end, we are developing the capability to analyze the past combat episodes off-line, and learn from obvious errors. In the above case, the Soar/IFOR agent records in its episodic memory that it got shot down. Its episodic memory of the combat also reveals that it never detected the opponent's missile firing behavior. Simultaneously, however, the episodic memory will note that the agent did face a mysterious maneuver that it was unable to track (corresponding to the missile firing with a 25-degree angle-off). Based on this episodic memory, the agent can learn that the human pilot can fire a missile from a 25-degree angle-off.

4 Practical Aspects of Using Chunking

Given the Soar/IFOR agents' real-world environment, the costs and benefits of chunking have to be evaluated from a practical perspective. The key question here is: Do the benefits of chunking outweigh its costs as it stands *today*? In this regard, the following factors need to be taken into account:

1. The Soar/IFOR agents' current knowledge is already encoded in a highly optimized form, so that they can rapidly respond to opponents' maneuvers. It is difficult for chunking to improve upon such decisions, other than to reorganize the encoded knowledge somewhat, as described above.

2. The agents' current knowledge is the result of extensive knowledge acquisition sessions. Some of the tactical knowledge gained from these sessions is highly sophisticated and a result of careful analysis of the capabilities of the opposing forces. It is difficult for chunking techniques to reconstruct, much less improve on, this expertise.

3. Chunks learned are sometimes highly specific—their conditions refer to the agent's current situation in terms of the value of its altitude, speed, range from an opponent, etc. Such chunks do not transfer (apply) to other similar situations, thus reducing the effectiveness of chunking.

4. The learning process itself can incur development overhead. Modifications to agent code can invalidate previously created chunks. Thus as the agents are modified, training sessions must be run repeatedly in order to produce an up-to-date set of chunks.

The above practical issues in applying chunking, combined with our earlier observations regarding the lack of overall speedups, implies that on-line chunking has to be very carefully applied, if at all, in service of speedups. We find it expedient to turn chunking on when the agents are making certain types of decisions, and turn it off elsewhere.

5 Long-Term Prospects

As development of Soar/IFOR proceeds, new opportunities continue to present themselves for making more extensive use of machine learning, and to employ existing learning abilities in new ways. Episodic memory is a good example of the latter: once an agent has the ability to remember previous episodes, a variety of possibilities for learning from those episodes present themselves. As the added capabilities afforded by machine learning accumulate, and the costs associated with learning are mitigated, the benefits stemming from learning are expected to dominate the costs to a greater and greater extent.

There is reason to believe, in fact, that eventually further improvement in performance of Soar/IFOR agents will only be achievable by means of machine learning. As long as the decision making of Soar/IFOR agents is governed by fixed rules, wily human opponents will learn ways of gaining advantages over the agents. This will be especially true if and when these agents are integrated into training devices that are used on a routine basis. If current work on enabling Soar/IFOR to learn from experience can be applied to a range of situations and scenarios, then human trainees will find simulations to be continually challenging, and able to put their tactical skills fully to the test.

Acknowledgement

We gratefully acknowledge Paul Rosenbloom's comments on this paper, as well as the contribution of the other members of the team involved in the creation of the Soar/IFOR agents, including John Laird, Randolph Jones, Karl Schwamb, and Frank Koss. This research was supported under contract N00014-92-K-2015 from the Advanced Systems Technology Office (ASTO) of the Advanced Research Projects Agency (ARPA) and the Naval Research Laboratory (NRL) to the University of Michigan, via a subcontract to USC; and under contract N66001-95-C-6013 from ARPA and the Navael Command and Ocean Surveillance Center, RDT&E division (NRAD).

References

[1] R.B. Calder, J.E. Smith, A.J. Courtemanche, J.M.F. Mar, and A.Z. Ceranowicz. ModSAF behavior simulation and control. In *Proceedings of the Third Conference on Computer Generated Forces and Behavioral Representation*, pages 347–359, Orlando, FL, March 1993. Institute for Simulation and Training, University of Central Florida.

[2] R.B. Doorenbos. Matching 100,000 learned rules. In *Proceedings of the National Conference on Artificial Intelligence*, pages 290–296, Menlo Park, CA, August 1993. AAAI.

[3] W.L. Johnson. Agents that learn to explain themselves. In *Proceedings of the National Conference on Artificial Intelligence*, pages 1257–1263, Seattle, WA, August 1994. AAAI, AAAI Press.

[4] J.E. Laird, A. Newell, and P.S. Rosenbloom. Soar: An architecture for general intelligence. *Artificial Intelligence*, 33:1–64, 1987.

[5] S. Minton. Quantitative results concerning the utility of explanation-based learning. *Artificial Intelligence*, 42(2–3):363–391, 1990.

[6] T. M. Mitchell, Keller R. M., and S. T. Kedar-Cabelli. Explanation-based generalization: A unifying view. *Machine Learning*, 1(1):47–80, 1986.

[7] P. S. Rosenbloom and J. E. Laird. Mapping explanation-based generalization onto soar. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 561–567, 1986.

[8] K.B. Schwamb, V.F. Koss, and D. Keirsey. Working with ModSAF: Interfaces for programs and users. In *Proceedings of the Fourth Conference on Computer Generated Forces and Behavior Representation*, pages 395–399, Orlando, FL, May 1994.

[9] V.J. Shute and J.W. Regian. Principles for evaluating intelligent tutoring systems. *Journal of Artificial Intelligence in Education*, 4(2/3):245–273, 1993.

[10] M. Tambe, W.L. Johnson, R.M. Jones, F. Koss, J.E. Laird, P.S. Rosenbloom, and K. Schwamb. Intelligent agents for interactive simulation environments. To appear in *AI Magazine*, Spring 1995.

[11] M. Tambe, A. Newell, and P. S. Rosenbloom. The problem of expensive chunks and its solution by restricting expressiveness. *Machine Learning*, 5(3):299–348, 1990.

Biographies

W. Lewis Johnson is a project leader at USC/ISI and a research assistant professor in the USC Department of Computer Science. Dr. Johnson received his A.B. degree in Linguistics in 1978 from Princeton University, and his M.Phil. and Ph.D. degrees in Computer Science from Yale University in 1980 and 1985, respectively.

Milind Tambe is a computer scientist at USC/ISI and a research assistant professor with the computer science department at USC. He completed his undergraduate education in computer science from the Birla Institute of Technology and Science, Pilani, India in 1986. He received his Ph.D. in 1991 from the School

of Computer Science at Carnegie Mellon University, where he continued as a research associate until 1993.