

Learning Procedural Planning Knowledge in Complex Environments

Douglas J. Pearson

Artificial Intelligence Laboratory
The University of Michigan, 1101 Beal Ave.
Ann Arbor, MI 48109, USA
dpearson@umich.edu, <http://ai.eecs.umich.edu>

Autonomous agents functioning in complex and rapidly changing environments can improve their task performance if they update and correct their world model over the life of the agent. Existing research on this problem can be divided into two classes. First, reinforcement learners that use weak inductive methods to directly modify an agent's procedural execution knowledge. These systems are robust in dynamic and complex environments but generally do not support planning or the pursuit of multiple goals and learn slowly as a result of their weak methods. In contrast, the second category, theory revision systems, learn declarative planning knowledge through stronger methods that use explicit reasoning to identify and correct errors in the agent's domain knowledge. However, these methods are generally only applicable to agents with instantaneous actions in fully sensed domains.

This research explores learning *procedural* planning knowledge through deliberate reasoning about the correctness of an agent's knowledge. As the system, IMPROV, uses a procedural knowledge representation it can efficiently be extended to complex actions that have duration and multiple conditional effects, taking it beyond the scope of traditional theory revision systems. Additionally, the deliberate reasoning about correctness leads to stronger, more directed learning, than is possible in reinforcement learners.

An IMPROV agent's planning knowledge is represented by production rules that encode preconditions and actions of operators. Plans are also procedurally represented as rule sets that efficiently guide the agent in making local decisions during execution. Learning occurs during plan execution whenever the agent's knowledge is insufficient to determine the next action to take. This is a weaker method than traditional plan monitoring, where incorrect predictions trigger the correction method, as prediction-based methods perform poorly in stochastic environments.

IMPROV's method for correcting domain knowledge is primarily based around correcting operator precon-

ditions. This is done by generating and executing alternative plans in decreasing order of expected likelihood of reaching the current goal. Once a successful plan has been discovered, IMPROV uses an inductive learning module to correct the preconditions of the operators used in the set of k plans (successes and failures). Each operator and whether it lead to success or failure is used as a training instance. This *k-incremental* learning is based on the last k instances and results in incremental performance which is required in domains that are time-critical. K-incremental learning is stronger than traditional reinforcement learning as the differences between successful plans and failed plans lead to better credit assignment in determining which operator(s) were incorrect in the failed plans and how the operator's planning knowledge was wrong.

Actions are corrected by recursively re-using the precondition correction method. The agent's domain knowledge is encoded as a *hierarchy* of operators of progressively smaller grain size. The most primitive operators manipulate only a single symbol, guaranteeing they have correct actions. Incorrect actions at higher levels are corrected by changing the preconditions of the sub-operators which implement them. For example, the effects of a brake operator are encoded as more primitive operators which modify the car's speed, tire condition etc. IMPROV's correction method is recursively employed to change the preconditions of these sub-operators and thereby correct the planning knowledge associated with the brake operator's actions. This method allows IMPROV to learn complex actions with durations and conditional effects.

The system has been tested on a robotic simulation and in driving a simulated car. We have demonstrated that k-incremental learning outperforms single instance incremental learning and that a procedural representation supports correcting complex non-instantaneous actions. We have also shown noise-tolerance, tolerance to a large evolving target domain theory and learning in time-constrained environments.