

Learning by Observation and Practice: A Framework for Automatic Acquisition of Planning Operators

Xuemei Wang *

School of Computer Science
Carnegie Mellon University
Pittsburgh PA 15213-3891
wxm@cs.cmu.edu

The knowledge engineering bottleneck is a central problem in the field of Artificial Intelligence. This work addresses this problem in the context of planning systems. It automatically learns planning operators by observing expert agents and by subsequent knowledge refinement in a learning-by-doing paradigm. Our learning method is implemented on top of the PRODIGY architecture (Carbonell *et al.* 1992).

The learning system is given at the outset the description language for the domain, which includes the types of objects and the predicates that describe states and operators. The observations of an expert agent consist of: 1) the sequence of actions being executed, 2) the state in which each action is executed (*pre-state*), and 3) the state resulting from the execution of each action (*post-state*). Our planning system uses STRIPS-like operators and the goal of this work is to learn the preconditions and the effects of the operators. We assume that the operators have conjunctive preconditions and no conditional effects, everything in the state is observable, and there is no noise in the state.

The architecture for learning by observation and practice in planning includes *the observation module, the learning module, the planning module, and the plan execution module*, as illustrated in Figure 1.

Operators for the domain are learned from these observation sequences in an incremental fashion utilizing a conservative specific-to-general inductive generalization process. When an operator is observed for the first time, the system creates the corresponding operator such that its precondition is the complete *pre-state* and its effect is the difference between the *post-state* and *pre-state*. Operators thus learned may have extra preconditions that correspond to the irrelevant features in the state. These extra preconditions are removed incrementally if they are not present in the *pre-states* of the new observations. In order to further refine the new operators to make them correct and complete, and to evaluate the new operators, the system uses them to solve practice problems. The system first generates an approximate plan to solve the practice problem using the par-

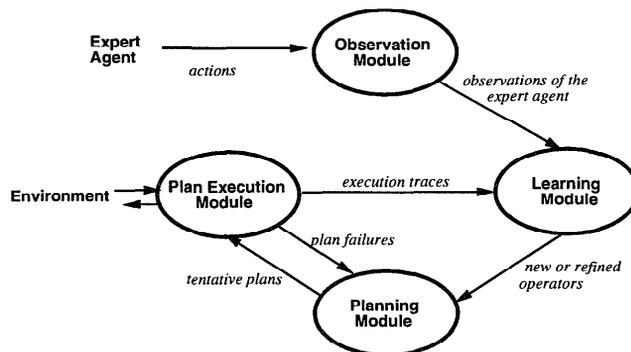


Figure 1: Overview of the data-flow among different modules of the learning system. The observation module provides the learning module with observations of the expert agent. The learning module formulates and refines operators in the domain. The planning module is essentially the PRODIGY planner modified to use some heuristics for planning with incomplete and incorrect operators as well as for plan repair, it generates tentative plans to solve practice problems. The plan execution module executes the plans, providing the learning module with the execution traces, and passing the plan failures to the planning module for plan repair.

tially incorrect and incomplete operators, then it executes the plan. When an operator fails to apply, it repairs the plan and continues execution, until the problem is solved or a resource bound is exceeded. The system also refines operators based on the executions. In summary, operators are refined through a process which integrates learning, planning, plan repair, and execution. See (Wang 1994) for more details of this research. This learning method has been partially demonstrated in the extended-strips domain and the process planning domain. We are currently performing more extensive tests in these domains. We plan to extend the algorithm to handle situations when part of the state is not observable, and there is noise in the state.

References

- Carbonell, J. G., and The PRODIGY Research Group. 1992. PRODIGY4.0: The manual and tutorial. TR CMU-CS-92-150, Carnegie Mellon University.
- Wang, X. 1994. Learning Planning Operators by Observation and Practice. In *Proceedings of the Second International Conference on AI Planning Systems*.

*Thanks to Jaime Carbonell for his suggestions and support in this work. This research is sponsored by the Wright Laboratory, Aeronautical Systems Center, Air Force Materiel Command, USAF, and the Advanced Research Projects Agency (ARPA) under grant number F33615-93-1-1330. Views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing official policies or endorsements, either expressed or implied, of Wright Laboratory or the United States Government.