A Model of Knowledge-Based Skill Acquisition

Stellan Ohlsson and Ernest Rees
Learning Research and Development Center
University of Pittsburgh
Pittsburgh, PA 15260, U. S. A.
stellan@vms.cis.pitt.edu

Abstract

We hypothesize that two important functions of declarative knowledge in learning is to enable the learner to detect and to correct errors. We describe psychologically plausible mechanisms for both functions. The mechanisms are implemented in a computational model which learns cognitive skills in three different domains, illustrating the cognitive function of abstract principles, concrete facts, and tutoring messages in skill acquisition.[1]

Practice and Knowledge

Practice consists of repeated attempts to solve problems which stretch the learner's competence. The paradox of practice is that the learner is deliberately setting out to solve a problem which he or she knows is beyond his or her current competence. It is far from obvious how this produces learning; and yet, there is no evidence that skills can be acquired without practice.

Beginning with the seminal papers by John R. Anderson and co-workers (e. g., Anderson & Kline, 1979) and by Anzai and Simon (1979), a computational interpretation of learning from practice has been developed over the past fifteen years. It can be summarized in three hypotheses:

(a) *The Weak Method Hypothesis.* When the learner is faced with a problem beyond his or her current competence, he or she uses weak problem solving methods such as analogical inference, forward search, or means-ends analysis to generate task relevant, but possibly inefficient, actions.

(b) *The Memory Storage Hypothesis.* Actions, even inefficient actions, generate information about the task environment, e. g., information about the effects of actions and about the properties of objects. This information is stored in memory.

(c) *The Skill Induction Hypothesis.* One or more learning mechanisms (composition, subgoaling, etc.) revise the current skill on the basis of the information in memory.

Repeated cycles through (a), (b) and (c) result in a domain-specific adaptation of the weak method which can solve the (class of) practice problem(s) efficiently. The three hypotheses can be articulated in different ways to generate a wide variety of specific learning models (see Klahr, Langley, & Neches, 1987).

[1] Preparation of this paper was supported by grant No. N00014-89-J-1681 from the Cognitive Science Program of the Office of Naval Research. The opinions expressed are not necessarily those of the funding agent and no endorsement should be inferred.

The empirical status of this mental bootstrapping theory of practice is still an open question, although its most dubious assumption--that people store large amounts of information in memory while engaged in the capacity demanding process of solving problems--has survived at least one attempt at falsification (Ohlsson, 1991).

The major limitation of the theory is that it depicts procedural knowledge as a closed system: Problem solving skills beget problem solving steps, which in turn beget new problem solving skills. There is no point along this loop at which prior knowledge about the task environment can influence the construction of the new skill. However, humans always learn in the context of prior domain knowledge.

A more complete theory of practice must describe how skill acquisition is influenced by at least three types of knowledge items: abstract principles, concrete facts, and tutoring messages.

Abstract principles are common in mathematics and science. The principle of one-one mapping is a simple example. It plays a crucial role in learning how to count a set of objects (Gelman & Gallistel, 1978). The laws of conservation of mass and energy are examples of principles in science.

Concrete facts are important in both technical domains and in everyday life. The fact that alcohol molecules are characterized by an OH-group is useful when constructing structural formulas in organic chemistry (Solomon, 1988).

Tutoring messages are short verbal instructions, uttered during practice. "Don't borrow unless the minuend digit is smaller than the subtrahend digit," uttered in the context of practice on subtraction with regrouping, is an example. One-on-one tutoring is a very efficient form of instruction (Bloom, 1984).

The purpose of this paper is to describe a computational model which embodies a unified view of the function of abstract principles, concrete facts, and tutoring messages in skill acquisition.

Learning as Error Correction

By necessity, a novice makes many errors while executing a skill; by definition, mastery is characerized by the absence of errors; hence, the gradual increase of competence during practice consists in the successive elimination of errors. Each error provides an opportunity to learn how to avoid similar errors in the future. To make use of such a learning opportunity, the learner must be able to (a) detect that an error as occurred, and (b) compute the appropriate revision of the current skill. We propose that the function of domain knowlege in skill acquisition is precisely to enable the learner to detect and to correct errors.

Learners can detect errors in three different ways: by self-monitoring, by observing the environmental effects, and by being told by others (Reason, 1990, Chap. 6). We focus on the first of these three methods. Learners monitor themselves, we suggest, by testing each new cognitive result (inference or knowledge state) for consistency with prior knowledge about the domain. For example, a Pittsburgh driver who is driving towards the river from the airport on the back roads and who sees a sign saying "route 60 south" recognizes that he or she has made an error, at least if he or she knows that the river is north of the airport. A chemistry student who gets more mass out of an experiment than he or she put in recognizes that an error was made, because this violates the law of conservation of mass. In each instance, detecting the error requires prior knowledge about the domain. During deliberate learning we constantly monitor the situations (problem states) we create for consistency with what we know about the domain and we recognize inconsistencies as errors. The more knowledge, the more powerful the self-monitoring ability.

Learners can correct an error, we suggest, by determining the conditions that produced it and then revising the current skill so as to prevent the relevant action from applying under those conditions. For example, the bewildered Pittsburgh driver might try to figure out which turn was wrong and correct his or her driving accordingly. The identification of the conditions that produced the error will result in a restriction on the relevant action, e. g., "remember not to turn right after exiting the parkway at Clairton."

In summary, according to our theory the learner monitors himself or herself by testing the consistency of each new conclusion or problem state with prior knowledge. Inconsistencies reveal errors which in turn trigger revisions which prevent those errors from occurring in the future. Over the course of practice, the errors are successively eliminated. The new skill has been mastered when no further errors occur.

A Simulation Model

A computational model that instantiates our theory must have (a) a performance component, (b) a representation for prior knowledge, (c) a mechanism for detecting errors, and (d) a mechanism for correcting errors. Our model is called the *Heuristic Searcher* (HS).

Performance component. HS is a vanilla flavored production system language. Rules have a goal and a conjunction of situation features in their left-hand sides and a single problem solving operator in their right-hand sides. Hence, each step in the problem space is controlled by a single production rule. There is no conflict resolution. If more than one rule fires, multiple new knowledge states are created. The system executes best-first search if supplied with an evaluation function by which to rank problem states and either depth-first or breadth-first search otherwise. HS is not an hypothesis about the human cognitive architecture. Our theoretical committment is limited to the two assumptions that cognitive skills are encoded as sets of production rules and that people at least sometimes solve unfamiliar problems through forward search. Both hypotheses are strongly supported by data (Anderson, in press; Newell & Simon, 1972).

Knowledge representation. Prior domain knowledge is encoded in datastructures called *state constraints*. Syntactically, a state constraint is an ordered pair <Cr, Cs>, where Cr and Cs are patterns, i. e., conjunctions of properties similar to the condition sides of production rules. The *relevance pattern* Cr has to match the current knowledge state for the constraint to be relevant and the *satisfaction pattern* Cs has to match for the constraint to be satisfied. States in which Cr match but Cs does not are called *constraint violations*. For example, a fact like "Fifth Avenue is one-way in the easterly direction" would be encoded as "if vehicle X is moving along Fifth Avenue, X had better be going east". Vehicles not moving along Fifth Avenue are not subject to the constraint; a vehicle going east on Fifth Avenue conforms with the constraint; a vehicle going west constitutes a constraint violation. Constraints are not inference rules or operators. They do not generate new conclusions or revise knowledge states. They test whether certain properties are true of the current knowledge state.

Error detection. When the current rule set generates a new knowledge-state, the latter is matched against all state constraints with the same pattern matcher that matches the rule conditions. Constraints in which Cr does *not* match are ignored, as are those in which *both* Cr and Cs match. Neither class of constraints warrant any action on the part of the system. Constraints for which Cr does match but Cs does not are recorded as violated. A constraint violation signals an inconsistency between the system's prior knowledge about the domain and the new outcome generated by the current rule set and it is interpreted as an error. HS assumes that the rule set is at fault.

Error correction. HS assigns blame to the last rule that fired, i. e., the rule that produced the violating knowledge state. A faulty rule is revised in two different ways. First, the relevance pattern by itself is regressed through the rule with a version of the standard goal regression algorithm (Nilsson, 1980) and the ***negation*** of the result added to the condition side of the rule. This produces a rule that only applies in situations in which the constraint is irrelevant. Second, the entire constraint is regressed through the rule and the result added to the rule condition (without negating it). This produces a rule which only applies in situations in which the constraint is ensured to be satisfied.

Curing a rule from violating one constraint does not garantee that the rule is correct; it might still violate other constraints. Multiple revisions of a rule are common in HS' learning. Because a skill consists of large number of rules, each of which might need multiple revisions, skill acquisition is necessarily gradual.

Three Applications

Three applications of HS have been implemented to date. They are summarized in Table 1. They illustrate that the model can learn from each of the three types of knowledge items specified previously: abstract principles, concrete facts, and tutoring messages.

Learning from abstract principles. Developmental data indicate that children construct the skill of counting a set of objects on the basis of (implicit) knowledge of abstract counting principles (Gelman & Gallistel, 1978). The main supporting phenomena are that children can transfer their counting routines to non-standard counting tasks and that they can evaluate counting performances that they cannot produce (Gelman & Gallistel, 1978; Gelman & Meck, 1986). The counting principles are abstract ideas like the one-one mapping principle. Expressed as a state constraint, this principle becomes "if object X has been assigned object Y, there should not be a third object Z assigned to either X or Y". The HS model learns the correct counting skill if given state constraints

Table 1. Three applications of the HS model.

Problem domain	Type of knowledge given to the model	Skill acquired by the model
Counting	Abstract principles, e. g. the one-one mapping principle.	To count a set of objects (see Ohlsson & Rees, 1991a).
Chemistry	Concrete facts, e. g. that alcohol molecules have OH-groups.	To derive the Lewis structure for a given molecular formula (see Ohlsson, in press-a).
Subtraction	Tutoring messages, e. g., "don't regroup unless the subtrahend is larger than the minuend."	Subtraction with regrouping (see Ohlsson, Ernst, & Rees, in press).

corresponding to the counting principles (Ohlsson & Rees, 1991a) and it can transfer the learned skill to other counting tasks (Ohlsson & Rees, 1991b).

Learning from concrete facts. The skill of constructing a Lewis structure on the basis of the molecular formula is a routine scientific skill taught at the begining of most organic chemistry courses. Textbooks teach this skill by first stating a general but weak procedure and then providing practice problems (e. g., Solomons, 1988). The general procedure is inefficient and must be specialized to particular classes of molecules. We gave HS a version of the general skill and provided it with state constraints expressing facts about three classes of molecules (alcohols, hydrocarbons, and ethers). An example of a fact is that alcohols have an OH-group ("if this is an alcohol molecule, it had better have an OH-group"). The model learned specialized versions of the general procedure for each type of molecule and its learning exhibited the negatively accelerated curve typical of human skill acquisition; see (Ohlsson, in press-a) for a more detailed discussion of these results.

Learning from tutoring messages. Students typically need tutoring to acquire the correct procedures for place value arithmetic. We gave HS an initial rule set which could solve canonical subtraction problems, i. e., problems in which the subtrahend digit is always smaller than the minuend digit in the same column. We then tutored the system through the learning of the regrouping procedure. The state constraints encoded typical tutoring messages, e. g., "don't borrow unless the minuend digit is smaller than the subtrahend digit." The predictions from this simulation experiment contradicted the current wisdom that regrouping is easier to learn than alternative subtraction methods; see Ohlsson (in press-b) and Ohlsson, Ernst & Rees (in press) for a detailed discussion of the results.

Discussion

Skill acquisition always occurs within the context of the learner's prior knowledge about the domain. Models of skill acquisition must explain the interaction between prior knowledge and problem solving experience during practice. We suggest that the cognitive function of domain knowledge is to enable the learner to monitor his or her own performance. The more domain knowledge he or she has, the better he or she can detect and correct errors.

The simulation model we built around this hypothesis learns in three different domains which supports the sufficiency and the generality of the learning mechanism. The model suggests new perspectives on three traditional problems in the theory of procedural learning. First, it predicts negatively accelerated learning curves, because the number of learning opportunities per practice trial will decrease as more and more errors are corrected. Second, it predicts low transfer of training between domains, because generality resides in the declarative knowledge and not in the skill itself. Finally, the model is consistent with the fact that one-on-one tutoring is the most efficient form of instruction, because tutors operate by helping the learner with the two main functions postulated in the model, i. e., to detect and correct errors.

The empirical validity of a complex simulation model is difficult to assess. The derivation of quantitative predictions from a computational model is tricky, because the model's behavior is determined not only by the hypotheses behind it but also by implementation details. Also, different models are seldom applied to the same phenomena, due to differences in the interests of their creators, making comparative evaluations difficult. No strong claims for the empirical validity of HS can be made at this time.

The theory behind HS is similar in spirit to the theory proposed by Schank (1986). According to the latter, people understand events by generating expectations from their current knowledge and they revise their knowledge when their expectations fail. Expectation failures and constraint violations are obviously similar types of events. Schank's theory is focussed on the understanding of other agents' actions rather than on problem solving and it represents knowledge in explanation patterns instead of rules, but the two theories share the hypothesis that learning is a response to an inconsistency between a cognitive outcome and existing knowledge.

This hypothesis might ultimately be undermined by empirical data. However, the problem of the interaction between prior knowledge and experience during practice will not go away. It must be solved before we can claim to fully understand skill acquisition.

References

Anderson, J. R. (in press). *Rules of the mind.* Hillsdale, NJ: Erlbaum.

Anderson, J. R., & Kline, P. J. (1979). A learning system and its psychological implications. *Proceedings of the Sixth International Joint Conference on Artificial Intelligence* (pp. 16-21). Tokyo, Japan.

Anzai, Y., & Simon, H. A. (1979) The theory of learning by doing. *Psychological Review, 86,* 124-140.

Bloom, B. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher, 13,* 4-16.

Gelman, R., & Gallistel, C. R. (1978). *The child's understanding of number.* Cambridge, MA: Havard University Press.

Gelman, R., & Meck, E. (1986). The notion of principle: The case of counting. In J. H. Hiebert, (Ed.), *Conceptual and procedural knowledge: The case of mathematics* (pp. 29-57). Hillsdale, NJ: Erlbaum.

Klahr, D., Langley, P., & R. Neches, (Eds.), (1987). *Production system models of learning and development* . Cambridge, MA: MIT Press.

Newell, A., and Simon, H. A. (1972). *Human problem solving.* Englewood Cliffs, NJ: Prentice-Hall.

Nilsson, N. (1980). *Principles of artificial intelligence.* Palo Alto, CA: Tioga.

Ohlsson, S. (1991). Memory for problem solving steps. *Program of the Thirteenth Annual Confernce of the Cognitive Science Society* (pp. 370-375). Hillsdale, NJ: Erlbaum.

Ohlsson, S. (in press-a). The interaction between knowledge and practice in the acquisition of cognitive skills. In A. Meyrowitz & S. Chipman, (Eds.), *Cognitive models of complex learning.* Boston: Kluwer.

Ohlsson, S. (in press-b). Artificial instruction: A method for relating learning theory to instructional design. In P. Winne & M. Jones, (Eds.), *Foundations and frontiers in instructional computing systems.* New York, NY: Springer-Verlag.

Ohlsson, S., Ernst, A. M., & Rees, E. (in press). The cognitive complexity of doing and learning arithmetic. *Journal of Research in Mathematics Eduation.*

Ohlsson, S., & Rees, E. (1991a). The function of conceptual understanding in the learning of arithmetic procedures. *Cognition & Instruction, 8,* 103-179.

Ohlsson, S., & Rees, E. (1991b). Adaptive search through constraint violation. *Journal of Experimental and Theoretical Artificial Intelligence, 3,* 33-42.

Reason, J. (1990). *Human error.* Cambridge, MA: Cambridge University Press.

Schank, R. (1986). *Explanation patterns.* Hillsdale, NJ: Erlbaum.

Solomons, T. W. G. (1988). *Organic chemistry* (4th ed.). New York, NY: Wiley.