In A. Segre (Ed.) Proceedings of the Sixth International Workshop on Machine Learning. Los Altos, CA: Morgan-Kaufman. 1989. pgs. 215-217

Discovering problem solving strategies: What humans do and machines don't (yet)

Kurt VanLehn
Departments of Computer Science and Psychology
Carnegie-Mellon University
Pittsburgh, PA 15213 U.S.A.
VANLEHN@PSY.CMU.EDU

An unusually detailed analysis and simulation of a human problem solving protocol uncovered 10 cases of strategies being discovered. Although most of these learning events were adequately modeled by existing machine learning techniques, several present interesting challenges for machine learning research. This paper briefly presents the experiment and the 10 learning events. The protocol analysis is detailed in VanLehn (1989). The simulation system, TETON, is described in VanLehn and Ball (in press).

The experiment and the protocol

Anzai and Simon (1979) published a Tower of Hanoi protocol that exhibits significant amounts of learning and has unusually clear verbal statements by the subject. Although Anzai and Simon modeled the main strategies and strategic transistions of the subject, they did not make a line-by-line comparison of the protocol to their model's behavior. The analysis presented here is a refinement of theirs. It uses the same nomenclature as Anzai and Simon (1979). The pegs of the Tower of Hanoi puzzle are labeled A, B and C, and the disks are numbered according to their size, with 1 being the smallest disk. The initial state of the puzzle has disks 1 through 5 on peg A. The goal is to get them all on peg C, subject to the constraints that a larger disk may never be placed on a smaller disk and only one disk may be moved at a time.

The following is a gloss of the protocol. During the first 30 minutes of the 90 minute protocol, the subject usually makes good moves. Some decisions, however, cause her difficulty apparently because she is looking ahead several moves in her mind's eye in order to evaluate alternative moves before choosing one. In a deliberate attempt to learn a better strategy, she embarks on a "experiment" that lasts about 45 minutes. She successively solves increasingly larger versions of the puzzle. She starts with the trivial puzzle that has just one disk on peg A, then solves the two-disk puzzle, and so on. Most of her learning occurs during this "experiment." She emerges with a clear strategy based on subgoaling with disks. For instance, in order to plan the initial move, she says (lines 110-114) "Of course, 5 will have to go to C, right? So, 4 will be at B. 3 will be at C. 2 will be at B. So 1 will go from A to C," then she makes her initial move, which is to move disk 1 to peg C. This disk-based subgoaling strategy is later supplanted by a pyramid-based strategy that follows the same recursive logic but uses pyramidally-shaped groups of disks in its calculations. The change from disks to pyramids is quite evident in the subject's choice of words (e.g., "The three blocking disks at B"). At the end of the protocol, the subject has, on my analysis, the rules shown in table 1. If a rule was acquired during the protocol, the table shows the line number of the rule's learning event in brackets after the rule. These rules will be discussed in numerical order.

Classification of the learning events

Unlike the other rules, rules 1 through 4 are applied on the very first opportunity that they can be applied. This suggests they may have been inferred from the puzzle's instructions or from common sense. For instance, rule 4 is an instantiation of the common sense idea that if you want to move something, and your move is blocked by an object, then move the blocking object out of the way.

The learning event for rule 5 is triggered by an impasse that occurs when the existing rules recommend moving disk 4 to peg B, but that move cannot be made legally. The subject uses rule 4 to deduce that this goal will always be a prerequisite for achieving the initial top level goal (moving disk 5 to peg C), so she adds 4-to-B as a top level goal. Thus, this learning event can be classified as impasse-driven explanation-based learning. Although it is a little unusual, because it occurs on an incorrect solution path that is abandoned soon after the learning event, this learning event could probably be handled by existing machine learning techniques.

1. Achieve the top level goals of the puzzle in the following order: get disk 5 to C, get disk 4 to C, get disk 3 to C, get disk 2 to C and get disk 1 to C.
2. Do not move the same disk on consecutive moves.
3. If there is a choice of where to put disk 1, and disk 2 is exposed, then put disk 1 on top of disk 2, thus creating a small pyramid.
4. If the goal is to move a given disk from a given peg to another given peg, and there is exactly one disk blocking the move, then get that blocking disk to the peg that is not involved in the move.
5. Before working on achieving any of the top level goals, get disk 4 to peg B. [12]
6. If the goal is to move a given disk from a given peg to another given peg, and the two-high pyramid is blocking the move, then get disk 1 to one of the two pegs involved in the move (thus allowing disk 2 to move out of the way of the move). [30-34]
7. If the goal is to move disk 2 from peg A to peg C, and disk 1 is on peg A, then move disk 1 to the peg that is not involved in the move. [78]
8. If the goal is to move disk N from peg A to peg C, and disk N-1 is on peg A, then get disk N-1 to the peg that is not involved in the move. [82]
9. If the goal is to move disk N from peg A to a given peg, and disk N-1 is on peg A, then get disk N-1 to the peg that is not involved in the move. [84]
10. If the goal is to move disk N from a given peg S to a given peg T, and disk N-1 is on S, then get disk N-1 to the peg that is not involved in the move. [99]
11. If the goal is to move a given disk from a given peg to another given peg, and disk D is the largest disk blocking the move, then get D to the peg that is not involved in the move.[121]
12. If the goal is to move a given pyramid from a given peg to another given peg, and pyramid P is the largest pyramid blocking the move, then get P on the peg that is not involved in the move.[179]

Table 1: Rules used by the subject during the protocol

Rule 6, which moves the two-high pyramid out of the way, is also learned at an impasse. The impasse occurs because the existing rules do not uniquely determine a move. The subject seems to perform a short look-ahead search in her imagination in order evaluate two alternative moves, then forms a rule that records what she has discovered. This type of reasoning does not fit the classic mold of explanation-based learning, for there is no sign of deduction from general rules. On the other hand, the reasoning is not much like similarity-based learning, for there is no induction over multiple exemplars. The reasoning seems to best fit a type of learning called *patching,* which was invented by Brown and VanLehn (1980) to explain how students acquire stable buggy strategies by encoding the results of applying a repair strategy to an impasse. In this case, patching led to the acquisition of a correct strategy rather than a buggy one. Patching is implemented by SIERRA (VanLehn, 1987).

The disk subgoaling strategy is acquired over a series of five learning events. The initial learning event is quite different from the others. It occurs at line 78, while the subject is reflecting on the solution she has just made to the two-disk puzzle. She seems to explain her move to herself, deducing from general principles that it is an appropriate strategy in the given circumstances. Thus, this learning event could be classified as explanation-based learning triggered by a deliberate plan of reflecting on the solution of a simpler version of the puzzle. Several machine learning programs (e.g., PRODIGY, Minton et al., 1987) use this type of learning. This learning event produces rule 7, the initial version of the disk subgoaling strategy.

The subsequent versions of the disk subgoaling strategy (rules 8 through 11) are learned by impasse-driven generalization. At exactly the points where an overly specific rule would fail to apply, the subject shows signs of impasses. There are four such occasions. In all cases, she generalizes the rule just enough to get it to match the situation present at the impasse. It takes four impasses to learn a fully general rule. This conservative, gradual generalization of the rule is a clear case of impasse-driven similarity-based learning. Several machine learning programs (e.g., SIERRA, VanLehn, 1987) implement this type of learning.

The pyramid strategy (rule 12) seems to appear in a fully general form, because subsequent applications of it do not cause impasses. Such one-trial learning is characteristic of explanation-based learning. However, explanation-based learning is not indicated in this case because the pyramid rule is not a deductive consequence of the existing rules. As Anzai and Simon suggested, it may be that the subject's learning is just the simple substitution of the perceptually more salient feature of "pyramid" for "disk" in the old disk subgoaling rule. This would make it similar to the perturbation-based learning of EURISKO (Lenat & Brown, 1984) and genetic algorithms (DeJong, 1988). However, the nature of protocol data makes it difficult to tell if this suggestion is correct.

Oddly, on a later use of the pyramid subgoaling rule, there is a second learning event. At line 197, the subject interrupts her use of the pyramid rule and starts using the old disk subgoaling rule. This suggests that she is deliberately comparing the two strategy's execution by running the disk subgoaling strategy overtly while covertly running the pyramid subgoaling strategy. To my knowledge, no machine learning program does this sort of checking, although it would be easy to implement.

A major feature of the protocol is the subject's "experiment" of successively solving larger puzzles in order to discover a better solution strategy. As evidence for the sophistication of her experiment, there are signs that she deliberately ignored rule 6 in order to find a more general rule. This was a fortunate choice, for rule 6, when used in combination with rules 1 through 5, suffices to solve any puzzle smaller than five disks. Had she not ignored rule 6, she may never have suffered the impasses that seem to be crucial for acquiring a general rule. To my knowledge, no machine learning program has demonstrated such sophistication in its approach to strategy acquisition.

The trigger for this extended learning event is not clear. There is no sign of pauses or confusion prior to the initiation of the experiment (lines 70-74). Instead, the experiment seems to be triggered by curiosity, for the subject says "I wonder if I've found something new..."

Conclusions

The overall picture one gets is that the subject is deliberately constructing a theory about Tower of Hanoi strategies. When she detects a deficiency in her theory, usually in the form of an impasse, she attempts to rectify it using deduction, experimentation, induction, or if all else fails, a repair strategy. She apparently has some "curiosity" demons preset to notice interesting events and propose an exploration of them. She seems to have set the noise threshold, so to speak, on her cognitive system in such a way that small perturbations are allowed to creep into the rules, which sometimes leads to unanticipated improvements. Clearly, there is no machine learning system on earth that includes all these styles of learning, and yet, there is nothing stopping us from building one.

References

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

DeJong, K. (1988). Learning with genetic algorithms: An overview. *Machine Learning, 3*(2/3), 121-138.

Lenat, D.B. & Brown, J.S. (1984). Why AM and Eurisko appear to work. *Artificial Intelligence, 23*(3), 269-294.

Minton, S., Carbonell, J.G., Etzioni, O., Knoblock, C. & Kuokka, D.R. (1987). Acquiring effective search control rules: Explanation-based learning in the Prodigy sytem. In P. Langley (Ed.), *Proceedings of the Fourth International Workshop on Machine Learning.* Los Altos, CA: Morgan Kaufmann.

VanLehn, K. (1987). Learning one subprocedure per lesson. *Artificial Intelligence, 31*(1), 1-40.

VanLehn, K. (1989). *Learning events in the discovery of problem solving strategies* (Tech. Rep. PCG-17). Dept. of Psychology, Carnegie-Mellon University.

VanLehn, K. & Ball, W. (in press, 19??). Teton: A large-grained architecture for studying learning. In VanLehn, K. (Ed.), *Architectures for Intelligence.* Hillsdale, NJ: Erlbaum.