In Proceedings of Cognitive Society Meeting, Bloomington, IN 1992.

873 - 878

A Fine-Grained Model of Skill Acquisition: Fitting Cascade to Individual Subjects

Randolph M. Jones & Kurt VanLehn
Learning Research and Development Center and
Computer Science Department
University of Pittsburgh
Pittsburgh, PA 15260
JONES@CS.PITT.EDU VANLEHN@CS.PITT.EDU

Abstract

The Cascade model of cognitive skill acquisition was developed to integrate a number of AI techniques and to account for psychological results on the *self-explanation effect*. In previous work, we compared Cascade's behavior to aggregate data collected from the protocols of 9 subjects in a self-explanation study. Here, we report the results of a fine-grained analysis, in which we matched Cascade's behavior to the individual protocols of each of the subjects. Our analyses demonstrate empirically that Cascade is a good model of subject behavior at the level of goals and inferences. It covers about 75% of the subjects' example-studying behavior and 60% to 90% of their problem-solving behavior. In addition, this research forced us to develop general feasible methods for matching a simulation to large protocols (approximately 3000 pages total). Finally, the analyses point out some weaknesses in the Cascade system and provide us with direction for future analyses of the model and data.

Introduction

Cascade is an integrated model of cognitive skill acquisition. It incorporates a number of methods from artificial intelligence, and was designed with attention to robust psychological findings. Elsewhere (VanLehn, Jones, & Chi, 1991, 1992), we have demonstrated that Cascade's mechanisms interact to account for the main qualitative findings involved in the self-explanation effect (Bielaczyc & Recker, 1991; Chi, Bassok, Lewis, Reimann, & Glaser, 1989; Chi, de Leeuw, Chiu, & LaVancher, 1991; Fergusson-Hessler & de Jong, 1990; Pirolli & Bielaczyc, 1989). In that research, we compared Cascade's behavior to aggregate data taken from the protocols of the 9 subjects in Chi et al.'s (1989) study.

In this paper, we refine the evaluation of Cascade by matching its behavior to the individual protocols of Chi et al.'s subjects. This research is similar to Newell and Simon's (1972) classic study on human problem solving, in that both attempt to determine how closely an AI program can simulate the protocols of individual subjects. However, there are three important differences between our study and Newell and Simon's. First, our task domain is physics, which is arguably much richer than the task domains they studied. Second, a considerable amount of learning occurs in the Chi et al. protocols. Third, the Chi et al. data consist of 252 protocols, each averaging 12 pages in length, so it would be infeasible to analyze them with problem-behavior graphs.

This work makes two important contributions. First, it provides evidence that Cascade's model of the subjects in Chi et al.'s study is quite accurate, even at the level of individual rules and goals. Second, it demonstrates a practical method for large scale comparisons of a simulation system to protocol data. We begin with an overview of the Cascade system. This is followed by a description of our paradigm for matching Cascade to the protocols and a brief discussion of our results. The paper concludes by describing implications of the results on future research with Cascade.

The Cascade system

Cascade is an AI system that integrates multiple strategies for problem solving and learning. Although the system has been applied to elementary probability and naive physics, the current analysis involves the domain of Newtonian physics, because this is the domain studied by the subjects. Due to space restrictions, we can only provide a summary of the system here. A detailed treatment can be found elsewhere (VanLehn & Jones, in press-a; VanLehn, Jones, & Chi, 1992).

Problem solving mechanisms

Cascade's overall control structure is based on a backward-chaining theorem prover (similar to Prolog), but it distinguishes between *explaining examples* and *solving problems*. Problems are presented as a set of literals describing a physical situation and a list of quantities for which Cascade must find values. An example is a problem along with a solution that consists of a sequence of lines describing partial results that lead up to the answer to the problem. To explain an example, Cascade explains (proves) each line. Whereas in problem solving Cascade must find a value, V, for a sought quantity, Q, in explaining an example line Cascade must prove why Q has a given value, V. Explanation is simpler than ordinary problem solving because the provided values help control search. People rarely explain every detail of every line, so Cascade can also *accept* that the current quantity has the stated value, instead of explaining it.

As Cascade explains an example, it stores a trace of its explanation (useful for solving subsequent problems), so more explanation leads to a larger stored derivation.

During problem solving, Cascade attempts to use its rule-based knowledge to find a value for a sought quantity. If this fails, the system tries to use a form of *transformational analogy* (Carbonell, 1983). That is, the system retrieves an example that is similar to the current problem and looks for a line in the example that mentions the sought quantity. If possible, it uses such a line to determine a value for the quantity. As Carbonell also found, this type of reasoning often leads to incorrect results. However, it is a strategy that subjects exhibit quite often.

Learning mechanisms

Cascade also includes two learning mechanisms for improving its problem-solving behavior. First, as we have mentioned, Cascade stores a trace of its solution as it explains examples. Many of the problems are analogous to one or more of the examples. Therefore, when the system works on a problem, it first attempts to create an analogical mapping between the problem and any similar examples. Then, when Cascade needs to solve a particular goal, it checks whether an analogous goal appeared in the example. If so, the system determines how it solved the goal in the example and uses that action in the current problem. In general, this mechanism leads to less search because it implies a better ordering of the rules in memory. This is a symbol-level learning mechanism called *analogical search control* (Jones, in press).

The second mechanism learns at the knowledge level (Dietterich, 1986), and is called *explanation-based learning of correctness* (VanLehn, Ball, & Kowalski, 1990). Cascade's knowledge base contains a number of overly general rules that are not used for general problem solving. However, when the system reaches an impasse on a problem or example and decides that the impasse is due to missing knowledge, it can use the overly general rules to patch its knowledge and introduce new standard rules. This method of learning is similar to knowledge-level learning methods proposed by Schank (1986), Lewis (1988), Anderson (1990), and others.

Fitting to individual subjects

For each subject, we set Cascade's parameters in order to approximate the subject's initial knowledge and example-explaining behavior. We then ran Cascade on the examples and problems that the subject worked on, collected data from the run, and analyzed them in several ways in order to determine Cascade's empirical accuracy. First we describe the parameter fitting and then the results of our analyses.

Initializing Cascade's parameters

Cascade's model of the subjects includes two parameters: the subjects' knowledge just before they explain the examples, and the subjects' decisions about which pieces of the examples to accept without explanation. Each paramater will be discussed in turn.

The subjects acquired their initial knowledge by reading the first several chapters of the textbook and from their earlier studies of physics and mathematics. Because we have no access to their learning history, nor a detailed test of their initial knowledge, we must guess their initial knowledge. Cascade's initial knowledge base for each subject was a subset of a fixed "rule library." The rule library consisted of 110 rules, including rules from the textbook, common sense rules, rules that are learnable via overly general rules, and 3 buggy physics rules that some subjects appeared to have. One buggy rule applies $F = ma$ to any force and not just a net force. Another asserts that the mass of a body is equal to its weight. The third assumes that the sign of all vector projections is positive.

There is no easy way to determine what a subject's initial knowledge is, but we made the best approximation we could by looking for rule use throughout each subject's entire set of protocols. As we found later, we sometimes made mistakes in selecting the initial knowledge. In these cases, we need to fix the mistakes, rerun the simulations

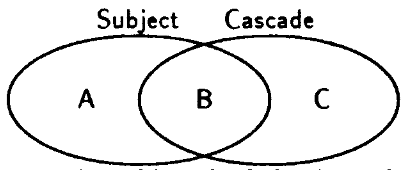

Figure 1. Matching the behaviors of Cascade and a subject

and redo our analyses. However, this will require months of work, so for now, we report the analyses with our imperfect choices of initial knowledge left intact. There were only a few of these cases, so we don't feel that the qualitative nature of our results will change.

The second parameter concerns how deeply the subjects explained the examples. When studying examples, subjects choose to explain some lines but not others. Even when they do explain a line, they may explain it only down to a certain level of detail and decide to take the example's word for the rest. For example, they might explain most of the line, $F_{a_x} = -F_a \cos(30)$, but not bother to explain where the minus sign comes from. Cascade does not model how the subjects decide which lines to explain and how deeply to explain them, so it must be told explicitly which sections to explain. Therefore, whenever Cascade is about to explain the the proposition, $Q = V$, it first checks to see if the literal `accept(Q,V)` is in the example's description. If the system finds such a literal, it merely accepts that Q's value is V without attempting to explain it.

We added `accept` propositions to Cascade by inspecting the subject's example protocols. If the subject merely read a line and said nothing else about it, then we entered an accept literal for the whole line. If the subject omitted discussion of a detail in a line, then we only accepted that detail, allowing Cascade to explain other goals involved in the line. In this fashion, the protocol data completely determined which lines and parts of lines Cascade explained.

The fit between Cascade and individual subjects

We are interested in two types of comparisons between the model and subject data. Suppose the diagram in Figure 1 represents the behaviors of a particular subject and Cascade's model of that subject. Region A represents subject behavior that Cascade failed to match. Region B represents the behavior that Cascade and the subject have in common. Region C represents Cascade behavior that the subject did not exhibit. The two comparisons we want are the ratio of region B behavior to Cascade's total behavior (regions B and C) and to the subject's total behavior (regions A and B). Table 1 shows the specific analyses conducted and their types.

Table 1. Analyses of Cascade's simulation of individual subjects.

1. How many of Cascade's example studying inferences were also made by the subject (BC vs. B)?
2. How many of the subject's example studying inferences were also made by Cascade (AB vs. B)?
3. How many of Cascade's problem solving inferences were also made by the subject (BC vs. B)?
4. How many of the subject's problem solving inferences were also made by Cascade (AB vs. B)?
5. Do the search control decisions made by the subject match those made by Cascade (AB vs. BC)?

In order to carry out these five analyses, we needed a way to quantify behaviors, which implies choosing a unit of analysis. This was not hard for matching region B to Cascade's behavior, because Cascade's behavior is well defined and explicit. For analyses 1 and 3, we used goals as the unit of analysis. After running Cascade, we classified each of its goals depending on the type of action Cascade took at that point. When explaining examples (analysis 1), these actions included deductively explaining the goal, accepting the goal without attempting to explain it, and encountering an impasse and learning a new rule. For problem solving (analysis 3), the actions included regular rule-based problem solving, regular use of transformational analogy, forced use of transformational analogy (for cases where the subject used transformational analogy but Cascade's normal control structure would have used regular problem solving), and encountering an impasse and learning a new rule. After classifying Cascade's behaviors, we determined what the subject's behavior was at each goal. We used the same classifications for subject behavior, but included the possibility of having an impasse and *not* learning a new rule (because the impasse was never resolved).

It was not as easy to determine a unit of analysis for matching region B to the subjects' behavior, so we used a variety of units, depending on the type of analysis being conducted. For analysis 2, we extended an earlier encoding of inferences made by subjects while explaining examples (Chi & VanLehn, 1991) and compared those to the inferences made by Cascade. For analysis 4, we coded a sample of the protocols at the level of Cascade-like goals. These goals are at the same grain size as

Cascade's goals, so the comparison is direct.

Because we are more interested in Cascade's simulation of the subjects' acquisition of physics rules than in its simulation of the chronology of their reasoning, analyses 1–4 ignored the order in which Cascade and the subject made inferences. Both Cascade's behavior and the subject's behavior were reduced to sets of inferences. We simply calculated the intersections and differences between the sets, just as shown in Figure 1. However, we cannot entirely ignore the chronology of inferences, because an earlier study indicated that analogical search control affects the location of impasses, which in turn determines what can be learned during problem solving (VanLehn & Jones, in press-a). Therefore, we used analysis 5 to determine whether subjects' rule choices during problem solving could be predicted by analogical search control.

Results of the analyses

Unfortunately, there is not enough space here to present the simulation runs and analyses in detail, so we will present a general summary and conclusions from the analyses. The details are presented elsewhere (VanLehn & Jones, in press-b).

Results on example explaining. We found that 95% of the example-explaining behavior generated by Cascade was matched by the subjects' behavior (analysis 1). This is not surprising because most of Cascade's example-studying behavior is determined by the parameter settings.

In analysis 2, we found that Cascade successfully accounted for 63% of the 227 explanation episodes in the subjects' example-studying protocols. Of the unmatched explanations, 61 were concerned with cognitive skills that we are not interested in modeling, such as algebraic equation solving. That left only 23 explanations (10% of the 227 total explanations) that Cascade should have been able to model. These fell into two groups: incorrect explanations (14 cases) and general comments (9 cases). The incorrect explanations indicate that Cascade needs more buggy rules than it currently has. In particular, many of the missing rules contained misconceptions about the relationship between acceleration and motion. The general comments indicate that the subjects have an ability to break out of Cascade's strict backward-chaining control structure and do plan recognition or mental modeling. These are certainly interesting and important cognitive skills, but we were surprised that they were used so rarely in this study. When we began developing Cascade, we expected plan recognition to be the most important kind of explanation. This analysis indicates that it occurs rarely and probably has little influence on subsequent problem solving. Overall, Cascade fails to model only 23 (14%) of the 166 explanation episodes that are relevant to the task domain, and we are encouraged by this result.

Results on problem solving. In analysis 3, we found that 97% of the 3947 goals generated by Cascade during problem solving were handled in the same way by the subjects. Of the 118 episodes that weren't matched by the subjects, most (98) involved transformational analogy. We were surprised by the prevalence of transformational analogy during problem solving, although it was certainly due in part to the fact that 12 of the 21 problems in the study were isomorphic (or nearly so) to one of the three examples.

Cascade's model of transformational analogy is too simple to describe adequately all the ways that transformational analogy was used by the subjects. A large number of the 98 cases occurred when subjects used a force diagram from an example to aid in drawing the force diagram for a problem. Cascade currently represents force diagrams in its standard equation-based representation, whereas the subjects were almost certainly using some type of visual representation. This partially explains why Cascade's transformational analogy fails in these cases.

Analysis 4 was quite time consuming, so we were only able to examine a small sample. Of the 225 total problem-solving protocols, we selected 4 that we thought were representative of the variety of approaches used by the subjects. Two protocols were from "good" problem solvers who got correct answers and two were from "poor" problems solvers who got incorrect answers. In addition, each of the pairs included a protocol that used mostly transformational analogy and a protocol that used mostly regular rule-based problem solving. This sample is clearly much too small, but it is a start. In the four protocols, we counted 151 total goals or inferences, excluding trivial arithmetic and algebraic goals. We found 15 cases in this analysis that the current implementation of Cascade failed to account for, so 90% of the subjects' problem-solving behavior is matched by the Cascade model. After several years of experience with these protocols, we feel intuitively that this figure is too high, and that a larger sample might yield a match that could be as low as 60%.

Results on search control. The first four analyses concentrated on matching the knowledge content of Cascade and the subjects without paying attention to *when* the knowledge is used. The order of inferences is determined by Cascade's control structure (backwards chaining) and its mechanism for choosing which rule to try first for achieving a goal (analogical search control, or if no analogical advice is available, then a default ordering of rules). As part of analysis 4, we fit the 151 subject goals to a backward-chaining control structure. Only 3 goals could not be fit, indicating that subjects occasionally make opportunistic inferences about the current situation that are not directly relevant to the current goal.

In order to evaluate Cascade's policy for choosing rules to apply to the current goal, we matched its choices for all 3947 goals to the choices of the subjects, and they agreed in 97.7% of the cases. In short, Cascade's simple control regime turned out to be a fairly good predictor of the order in which subjects make inferences.

Discussion

One contribution of this work is that it demonstrates a method for comparing large-scale AI simulations with protocols. Our general method consists of comparing the amount of shared behavior between the simulation and the subjects to the total simulation behavior and the total subject behavior. The unit of analysis for matching simulation behavior is straightforward, because Cascade's behavior is explicit for each goal it considers when explaining examples or solving problems. For matching subject behavior, we used two separate measures. In analysis 2 (explaining examples), we coded the subject protocols at the level of individual physics or math explanations, and compared the inferences with Cascade's. In analysis 4 (solving problems), we undertook a much more ambitious method, coding the protocols at the level of Cascade-like goals. This analysis allowed us to match the subjects' behavior to Cascade goal by goal, noting the locations where Cascade's model diverged from the subjects' behavior. Although rather time-consuming, our success with this type of encoding encourages us to continue the analysis with a larger sample of protocols.

The second contribution of this research is an empirical evaluation of Cascade's ability to model the behavior of individual subjects at a fine grain size. We discovered that Cascade can explain most of the subjects' example-studying and problem-solving behavior with its three major performance mechanisms: deduction, simple acceptance of example statements, and transformational analogy. Analyses 1–4 indicate that these three processes cover about 75% of the example studying behavior and 60–90% of the problem solving behavior. In addition, the behavior they do not cover mostly involves mathematical manipulations or other types of cognition that are outside the domain of study.

Finally, analysis 5 demonstrates that Cascade rather accurately models the subjects' overall control structure and local control choices. We were pleasantly surprised by this result, because we did not concentrate on these aspects during the system's development.

To put these results in perspective, we look at two other attempts to match cognitive models to individual subjects. Newell and Simon (1972) used GPS to match 80% of an indidividual subjects' behavior on cryptarithmetic problems. VanLehn's (1991) model for strategy discovery accounted for 96% of the behavior of a subject solving the "tower of Hanoi" problem. It is important to note that both of these studies involved modeling the behavior of a single subject. We used Cascade to model the behavior of several individuals, which is almost guaranteed to reduce the model's overall accuracy. With this in mind, Cascade's account of human behavior compares well with the older models.

Perhaps the most important benefit of this research is that it has shown us where some of Cascade's weaknesses are, and it has pointed out some more aspects of the data that should also be analyzed. For example, we found that Cascade's simple model of transformational analogy is inadequate. Subjects were quite clever at forming useful analogies with the examples, and especially their force diagrams. In addition, we were surprised to find that there were so few clear-cut cases of impasse-driven learning in the protocols. During analyses 1 and 3, we found that subjects only showed signs of impasses at 18 of the 44 times that Cascade encountered an impasse and used explanation-based learning of correctness to get out of it. Our initial hypothesis is that these events arise either from ingenious use of transformational analogy by the subjects, or they were actual impasses that were simply not verbalized in the protocols. Our future analyses will concentrate on these learning aspects and should tell us exactly why there were so few clear cases of learning.

Acknowledgements

This research was supported in part by contract N00014-88-K-0086 from the Cognitive Science division and contract N00014-86-K-0678 from the Information Sciences division of the Office of Naval Research.

References

Anderson, J. R. (1990). *Adaptive control of thought.* Hillsdale, NJ: Lawrence Erlbaum.

Bielaczyc, K., & Recker, M. M. (1991). Learning to learn: The implications of strategy instruction in computer programming. In L. Birnbaum (Ed.), *The International Conference on the Learning Sciences.* Charlottesville, VA: Association for the Advancement of Computing in Education.

Carbonell, J. G. (1983). Learning by analogy: Formulating and generalizing plans from past experience. In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell (Eds.), *Machine learning: An artificial intelligence approach.* Los Altos, CA: Morgan Kaufmann.

Chi, M. T. H., Bassok, M., Lewis, M. W.,

Reimann, P., & Glaser, R. (1989).

Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science, 13*, 145–182.

Chi, M. T. H., de Leeuw, N., Chiu, M., & LaVancher, C. (1991). *The use of self-explanations as a learning tool.* Manuscript submitted for publication.

Chi, M. T. H., & VanLehn, K. (1991). The content of self-explanations. *Journal of the Learning Sciences, 1*, 69–106.

Dietterich, T. G. (1986). Learning at the knowledge level. *Machine Learning, 1*, 287–316.

Ferguson-Hesler, M. G. M. & De Jong, T. (1990). Studying physics texts: Differences in study processes between good and poor solvers. *Cognition and Instruction,* 7, 41–54.

Jones, R. M. (in press). Problem solving via analogical retrieval and analogical search control. In S. Chipman & A. Meyrowitz (Eds.), *Machine learning: Induction, analogy, and discovery.* Boston: Kluwer Academic.

Lewis, C. (1988). Why and how to learn why: Analysis-based generalization of procedures. *Cognitive Science, 12*, 211–256.

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

Pirolli, P., & Bielaczyc, K. (1989). Empirical analyses of self-explanation and transfer in learning to program. *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society.* Hillsdale, NJ: Lawrence Erlbaum.

Schank, R. C. (1986). *Explanation patterns: Understanding mechanically and creatively.* Hillsdale, NJ: Lawrence Erlbaum.

VanLehn, K. (1991). Rule acquisition events in the discovery of problem solving strategies. *Cognitive Science, 15*, 1–47.

VanLehn, K., Ball, W., & Kowalski, B. (1990). Explanation-based learning of correctness: Towards a model of the self-explanation effect. *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society.* Hillsdale, NJ: Lawrence Erlbaum.

VanLehn, K., & Jones, R. M. (in press-a). Integration of analogical search control and

explanation-based learning of correctness. In S. Minton & P. Langley (Eds.), *Planning, scheduling and learning.* Los Altos, CA: Morgan Kaufmann.

VanLehn, K., & Jones, R. M. (in press-b).

Learning by explaining examples to oneself: A computational model. In S. Chipman & A. Meyrowitz (Eds.), *Cognitive models of complex learning.* Boston: Kluwer Academic.

VanLehn, K., Jones, R. M., & Chi, M. T. H. (1991). Modeling the self-explanation effect with Cascade 3. *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society.* Hillsdale, NJ: Lawrence Erlbaum.

VanLehn, K., Jones, R. M., & Chi, M. T. H. (1992). A model of the self-explanation effect. *Journal of the Learning Sciences, 2* 1–59.