Integrating Perceptual Representation Learning and Skill Learning in a Simulated Student

Nan Li and William W. Cohen and Kenneth R. Koedinger
School of Computer Science, Carnegie Mellon University
5000 Forbes Ave., Pittsburgh PA 15213 USA
Email: nli1@cs.cmu.edu, wcohen@cs.cmu.edu, koedinger@cmu.edu

Abstract—**One of the fundamental goals of artificial intelligence is to understand and develop intelligent agents that simulate human-level intelligence. This fundamental goal complements another essential goal in education, improving understanding of how humans acquire knowledge and how students may vary in their abilities to learn. Contributing to both goals, a lot of efforts have been made to develop intelligent agents that simulate human learning of math and science. However, constructing such a learning agent currently requires manual encoding of prior domain knowledge, which is both inefficient and less cognitively plausible. Previous cognitive science research has shown that one of the key factors that differentiates experts and novices is their different representations of knowledge. Moreover, for many existing learning algorithms, "better" representations often lead to more effective learning. We [1] recently proposed an efficient algorithm that acquires representation knowledge in the form of "deep features". In this paper, we integrate this algorithm into a simulated student, SimStudent, which learns procedural knowledge from example solutions and problem solving experience. We show that with the integration, prior knowledge engineering effort is reduced, learning performance is as good or better, and SimStudent becomes a more plausible simulation of human learning.**

I. Introduction

One of the fundamental goals of artificial intelligence is to understand and develop intelligent agents that simulate human-like intelligence. A large amount of effort (e.g., [2], [3]) has been put toward this challenging task. Further, education in the 21^{st} century will be increasingly about helping students not just to learn content but also to become better learners. Thus, we have a second goal of improving our understanding of how humans acquire knowledge and how students vary in their abilities to learn.

To contribute to both goals, considerable efforts (e.g., [4]) have been made to develop intelligent agents that model human learning of math, science, or a second language. Although such agents produce intelligent behavior with less human knowledge engineering than before, there remains a non-trivial element of knowledge engineering in the encoding of the prior domain knowledge (e.g., programming how to extract a coefficient from a term). This requirement increases the difficulty of constructing an intelligent agent. It also reduces the cognitive plausibility of the constructed agent, as human students entering a course do not necessarily have substantial domain-specific or domain-relevant prior knowledge. An intelligent agent that requires only domain-independent prior knowledge as given would be a great improvement.

Previous work in cognitive science (e.g., [5]) showed that one of the key factors that differentiates experts and novices is their different prior knowledge of world state representation. Experts view the world in terms of deep functional features (e.g., coefficient and constant in algebra), while novices only view it in terms of shallow perceptual features (e.g., integer in an expression). Recent work on perceptual expertise [6] and on vision in robotics (e.g., [7]) has shown the importance of perceptual representation learning across domains. We have recently developed a learning algorithm that acquires deep features automatically with only domain-independent knowledge as input [1]. In this paper, we integrate this representation learner into a machine-learning agent, SimStudent [4], and evaluate the proposed approach in three domains: fraction addition, equation solving, and stoichiometry [8].

II. A Brief Review of SimStudent

SimStudent is an intelligent agent that inductively learns skills to solve problems from demonstrated solutions and from problem solving experience. It is an extension of programming by demonstration [9] using inductive logic programming [10] as an underlying learning technique.

This skill knowledge is represented as production rules. The left side of Figure 1 shows an example of a learned production rule in its readable format[1]. The perceptual information part is acquired by the "where" learner. The precondition part is learned by the "when" learner. The operator function sequence part is created by the "how" learner. The rule to "divide both sides of $-3x=6$ by -3" shown at the left side of Figure 1 would be read as "given a left-hand side (i.e., $-3x$) and a right-hand side (6) of the equation, when the left-hand side does not have a constant term, then get the coefficient of the term at the left-hand side and divide both sides by the coefficient."

Note that operator functions are divided into two groups, domain-independent operator functions and domain-specific operator functions. Domain-independent operator functions are basic skills used across multiple domains. Hence, we assume that real students usually have knowledge of these simple skills prior to class. Domain-specific operator functions, on the other hand, are more complicated skills. Performing such operator functions usually require domain expert knowledge, which real students may not have.

[1]Actual production rules follow the LISP format.

- Original:
- Skill divide (e.g. -3x = 6)
- Perceptual information:
  - Left side (-3x)
  - Right side (6)
- Precondition:
  - Left side (-3x) does not have constant term
- Operator sequence:
  - Get coefficient (-3) of left side (-3x)
  - Divide both sides with the coefficient (-3)

- Extended:
- Skill divide (e.g. -3x = 6)
- Perceptual information:
  - Left side (**-3**, -3x)
  - Right side (6)
- Precondition:
  - ~~Left side (-3x) does not have constant term~~
  - -3 is the left child of the left side (-3x)
  - -3 is a signed number
- Operator sequence:
  - ~~Get coefficient (-3) of left side (-3x)~~
  - Divide both sides with the coefficient (**-3**)

Fig. 1. Original and extended production rules for divide in a readable format.

III. Integrating Representation Learning into SimStudent

Having reviewed SimStudent, we move to a discussion of representation knowledge acquisition as deep feature learning. As mentioned above, representation learning is important both for human knowledge acquisition, and in achieving effective machine learning. Missing deep feature knowledge sometimes causes real students to make errors in learning. We [1] examined the nature of deep feature learning in algebra equation solving, and discovered that it could be modeled as a grammar induction problem given observational data (e.g. equations in algebra). Expressions can be formulated as a context free grammar. The deep feature "coefficient" is a non-terminal symbol in one of the grammar rules. The perspective of viewing representation learning tasks as grammar induction problems also explains the cause of student errors.

Given the promising results, we believe the representation learner is effective in acquiring representation knowledge, and is a good model of real students. To better evaluate how the representation learner could affect the performance of an intelligent agent, we present how to integrate representation learning into SimStudent. As we have mentioned above, SimStudent is able to acquire production rules in solving complicated problems, but requires a set of domain-specific operator functions given as prior knowledge. In order to both reduce the amount of prior knowledge engineering needed for SimStudent and to build a better model of real students, we present a novel approach that integrates the representation learner into SimStudent. Figure 1 shows a comparison between production rules acquired by the original and the extended SimStudents. As we can see, the coefficient of the left-hand side (i.e., *-3*) is included in the perceptual information part in the extended production rule. Therefore, the operator function sequence no longer needs the domain-specific operator, *(coefficient -3x)*.

IV. Experimental Study

In order to evaluate whether the extended SimStudent is able to acquire correct knowledge with reduced prior knowledge engineering, we carried out an experiment in three domains: fraction addition, equation solving, and stoichiometry. Although not shown here, we have also demonstrated that the extended SimStudent can be used to discover models of human students that are better than those found by experts [11].

For each domain, the representation learner was first trained on a sequence of feature learning tasks. Then, SimStudent was tutored by an automatic tutor, which simulates the automatic tutor used by human students in some classroom study. All of the problems were also extracted from the same study.

We evaluated the effectiveness of SimStudent with two measurements: the amount of knowledge engineering needed, and the speed of learning. Experimental results show that, in all three domains, the original SimStudent given domain-specific operator functions required more than twice as much coding compared to the extended SimStudent given only domain-general operator functions. In addition, after trained on all problems, comparing with the original SimStudent with domain-specific operator functions , the extended SimStudent performed slightly better in equation solving, and significantly better in fraction addition and stoichiometry ($p < 0.0001$).

V. Concluding Remarks

Building an intelligent agent that simulates human-level learning is an essential task in AI and education, but building such systems often requires manual encoding of prior domain knowledge. We proposed a novel approach that integrates a representation learning algorithm into an intelligent agent, SimStudent, as an extension of the perception module. After the integration, the extended SimStudent is able to achieve better or at least comparable performance without requiring any domain-specific operator function as input.

References

[1] N. Li, W. W. Cohen, and K. R. Koedinger, "A computational model of accelerated future learning through feature recognition," in *Proceedings of 10th International Conference on Intelligent Tutoring Systems*, 2010, pp. 368–370.

[2] J. E. Laird, A. Newell, and P. S. Rosenbloom, "Soar: an architecture for general intelligence," *Artificial Intelligence*, vol. 33, no. 1, pp. 1–64, 1987.

[3] J. R. Anderson, *Rules of the Mind*. Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1993.

[4] N. Matsuda, A. Lee, W. W. Cohen, and K. R. Koedinger, "A computational model of how learner errors arise from weak prior knowledge," in *Proceedings of Conference of the Cognitive Science Society*, 2009.

[5] M. T. H. Chi, P. J. Feltovich, and R. Glaser, "Categorization and representation of physics problems by experts and novices," *Cognitive Science*, vol. 5, no. 2, pp. 121–152, June 1981.

[6] I. Gauthier, M. Tarr, and D. Bub, *Perceptual Expertise: Bridging Brain and Behavior*. New Your, NY: Oxford University Press, 2009.

[7] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, Washington, DC, 1999.

[8] N. Li, W. W. Cohen, and K. R. Koedinger, "Efficient cross-domain learning of complex skills," in *Proceedings of the 11th International Conference on Intelligent Tutoring Systems*, 2012.

[9] T. Lau and D. S. Weld, "Programming by demonstration: An inductive learning formulation," in *Proceedings of the 1999 international conference on intelligence user interfaces*, 1998, pp. 145–152.

[10] S. Muggleton and L. de Raedt, "Inductive logic programming: Theory and methods," *Journal of Logic Programming*, vol. 19, pp. 629–679, 1994.

[11] N. Li, W. W. Cohen, K. R. Koedinger, and N. Matsuda, "A machine learning approach for automatic student model discovery," in *Proceedings of the 4th International Conference on Educational Data Mining*, 2011, pp. 31–40.