

Automated Creation of Intelligent Tutoring to Support Personalized Online Learning

Nan Li, Abraham Schreiber, William W. Cohen, Kenneth R. Koedinger School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 nli1@cs.cmu.edu, abrahamjschreiber@gmail.com, wcohen@cs.cmu.edu, koedinger@cmu.edu

1 Introduction

With the fast development of modern technology, there has been a growing interest for leveraging online platforms for education in recent years such as Khan Academy, Stanford online courses and so on. In order to provide better online learning experiences, educators and researchers have worked intensively to develop personalized interactive tutoring systems that teach individual students according to their abilities, learning styles and so on. Nevertheless, building such tutoring systems requires artificial intelligence programming skills and cognitive psychology expertise.

SimStudent is a learning agent, which learns problem-solving skills by examples and by feedback on performance. It has been integrated into an existing authoring tool for Cognitive Tutors [1]. End-users can create intelligent tutoring systems easily by teaching SimStudent rather than by programming. SimStudent has been applied to various domains, which include multi-column addition, fraction addition, equation solving, stoichiometry, tic-tac-toe and so on. However, previous effort in constructing such intelligent learning agents (e.g., [2, 9]) currently requires manual encoding of prior domain knowledge, which is both time-consuming and error-prone. An intelligent agent that requires only domain-independent prior knowledge would be a great improvement both for the artificial intelligence community and for the education community.

2 Representation Learning and Automatic Feature Predicate Creation

Our idea was inspired by previous research in cognitive science. It has shown that one of the key factors that differentiates experts and novices is their different representations of knowledge (e.g., what is a fraction, what is a constant, what is a coefficient) [3]. Experts view the world in terms of deep functional features (e.g., coefficient and constant in algebra), while novices only view in terms of shallow perceptual features (e.g., integer in an expression). Learning deep features changes the representation on which future learning is based and, by doing so, improves future learning. However, how these deep features are acquired is not clear.

Therefore, we recently proposed an efficient algorithm that acquires representation knowledge in terms of “deep features” using grammar induction techniques. The input to the representation learner takes is unannotated/lightly annotated 1-dimensional/2-dimensional text. The learner acquires 1-dimensional/2-dimensional probabilistic context-free grammars that represent the underlying structure in the problems [6].

Furthermore, using domain-specific information encapsulated in the deep features of the grammar, we automatically generated a set of predicates that can be used by the intelligent agent to replace feature predicates that had to be manually constructed before [7].

3 Integrating Representation Learning and Skill Learning

In order to further evaluate how representation learning could affect the performance of an intelligent agent, we integrated this representation learning algorithm into SimStudent to provide a better representation. SimStudent is a machine-learning agent that inductively learns skills to solve problems from demonstrated solutions and from problem solving experience. It is an extension of programming by demonstration [5] using inductive logic programming [8] as an underlying learning technique. SimStudent's learning mechanisms include three parts: a retrieval path ("where") learner, a precondition ("when") learner, and a function sequence ("how") learner. Given a problem, the knowledge acquired by SimStudent defines "where" to look for useful information in the GUI, and "when" the useful information satisfies certain conditions, "how" to proceed.

The original SimStudent relies on a hand-engineered representation that encodes an expert representation given as prior knowledge. This limits its ability to model the state of prior knowledge of novice students, as novice students may not know such knowledge before to class. We replace the set of hand-engineered representations with the acquired representations, and further extend corresponding learning components to make better use of the acquired representation knowledge. Experiment results show that we reduce the amount of knowledge engineering effort required in constructing an intelligent agent, and get a better modeling of human learning.

In addition, the task of acquiring the precondition of the rule is a classification task. In SimStudent, we decoupled this learning module into two components, an unsupervised statistical module that learns the world representation and generates a set of feature predicates, and a supervised logic-based module that uses the generated feature predicates to acquire the precondition of the production rule. In comparison with this decoupled learning strategy, we adapt a joint model, deep belief network [4], to the precondition learning task. Experiment results show that deep belief network achieves reasonable performance (>80%), but is not as effective as the decoupled strategy (>90%).

4 Concluding Remarks

In this talk, we would like to summarize different components of our work, and give an overview of how the integration of representation learning and skill learning can assist intelligent agent construction. Moreover, we would like to discuss challenges in using existing machine learning techniques to build human-like intelligence. With a better model of human learning, we would further aid the automated creation of intelligent tutoring to support personalized online learning.

References

- [1] Vincent Aleven, Bruce M. McLaren, Jonathan Sewall, and Kenneth R. Koedinger. A new paradigm for intelligent tutoring systems: Example-tracing tutors. *International Journal of Artificial Intelligence in Education*, 19:105–154, April 2009.
- [2] Yuichiro Anzai and Herbert A Simon. The theory of learning by doing. *Psychological Review*, 86(2):124–140, 1979.

- [3] William G. Chase and Herbert A. Simon. Perception in chess. *Cognitive Psychology*, 4(1):55–81, January 1973.
- [4] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.
- [5] Tessa Lau and Daniel S. Weld. Programming by demonstration: An inductive learning formulation. In *Proceedings of the 1999 international conference on intelligence user interfaces*, pages 145–152, 1998.
- [6] Nan Li, William W. Cohen, and Kenneth R. Koedinger. Learning to perceive two-dimensional displays using probabilistic grammars. In *Proceedings of the 22nd European Conference on Machine Learning*, 2012.
- [7] Nan Li, Abraham Schreiber, William W. Cohen, and Kenneth R. Koedinger. Creating features from a learned grammar in a simulated student. In *Proceedings of the 20th European Conference on Artificial Intelligence*, 2012.
- [8] Stephen Muggleton and Luc de Raedt. Inductive logic programming: Theory and methods. *Journal of Logic Programming*, 19:629–679, 1994.
- [9] Kurt Vanlehn, Stellan Ohlsson, and Rod Nason. Applications of simulated students: an exploration. *Journal of Artificial Intelligence in Education*, 5:135–175, February 1994.