Advances in Cognitive Systems X (20XX) 1-6 Submitted X/20XX; published X/20XX

Apprentice Learner Architecture: A framework for modeling human learning from demonstrations and feedback in digital environments

Christopher James MacLellan CMACLELL@CS.CMU.EDU
Human-Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA

Abstract

Understanding the nature of human intelligence and developing intelligent agents capable of modeling humans are fundamental goals of cognitive systems research. Prior work modeling human problem solving has explored how hand-constructed domain models (e.g., production-rule models) can be used to explain human behavior. Typically, these models account for how humans improve their problem-solving performance given practice (i.e., speed-up learning), but they do not account for how humans acquire initial domain models. One approach that humans use to acquire knowledge in a new domain is apprenticeship learning, or learning from demonstrations and feedback from an expert. In the current work, I formalize the apprenticeship learning task for digital learning environments and present the Apprentice Learner Architecture, which provides a framework for building models of apprenticeship learning that align with this task formalization. Next, I briefly review how this model can be used to simulate and predicting human behavior in intelligent tutors. Finally, I conclude with directions for future work.

1. Introduction

One of the ultimate goals of cognitive systems research is to construct agents that are capable of human-level learning and problem solving. Prior research towards this goal has explored how researcher-authored models can be used to explain or emulate human behavior. For example, researchers using ACT-R (Anderson, 1993) account for human behavior by encoding psychological theories into cognitive models. These models embody the researchers' theories of the facts, skills, and goals that humans use when completing a given task. Researchers can then empirically test a theory through the comparison of the respective model's behavior with human behavior. Other researchers, using different architectures, such as SOAR and ICARUS (Langley, Laird, & Rogers, 2009), have taken a similar approach when attempting to mirror human-level capabilities. For example, TACAIR-SOAR (Jones, 1999), a hand-built cognitive model of combat fighter pilots, consists of over 5200 production rules that have been shown to generate believable human-like behavior during military training simulations. Key components of these systems include their abilities to solve novel problems using search and to improve their problem-solving performance given practice via techniques like hierarchical skill learning or reinforcement-learning.

While these systems possess the ability to learn from problem-solving experiences, they still require researchers to construct initial models of the facts, skills, and goals for each domain. For example, architectures such as ACT-R, SOAR, and ICARUS have capabilities for learning from

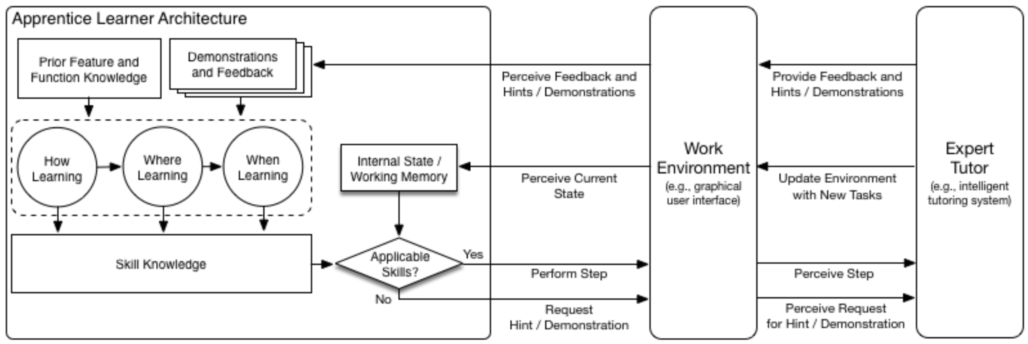

Figure 1: The Apprentice Learner Architecture and its interactions between the work environment and expert tutor. The architecture possesses three learning mechanisms (how, where, and when) to generalize demonstrations and feedback into skill knowledge that can be used for problem solving.

problem solving (Langley et al., 2009). However, for learning from problem solving to be successful these initial models must be complete in the sense that all novel problems encountered by an agent must be solvable using the initially provided cognitive model. This is often a difficult requirement to meet because it is difficult to know a priori all the knowledge that an agent will need to succeed.

In contrast, when a human is tasked with acquiring knowledge in a novel domain they rarely work alone to construct new domain knowledge through problem solving (assuming they have sufficient prerequisite knowledge). Instead, they often rely on existing domain experts to provide them with examples and feedback in the target domain. Students then reflect on these examples and feedback in order to learn new domain knowledge that can be used in subsequent problem solving. This kind of learning, which I refer to as apprenticeship learning, is different from learning through problem solving in that it is primarily about transferring expertise to the novice rather than the novice discovering the knowledge on their own.

In the current work I formalize apprentice learning as the task of learning new domain knowledge from expert demonstrations and feedback, and discuss how this task formalization aligns with the behavior of tutoring systems (Vanlehn, 2006), a type of digital learning environment that is designed to support human apprenticeship learning and that can provide data of human apprentice learning. Next, I present the Apprentice Learner Architecture, a framework for building models of apprenticeship learning within this formalization and discuss how it can be used to model human learning in a fraction arithmetic tutor. Finally, I discuss directions for future work.

2. Apprenticeship Learning

The apprentice learning task, which is depicted in Figure 1, consists of an agent (e.g., and apprentice learner agent or a human agent) learning from demonstrations and feedback on problem-solving attempts in a work environment. Ideally, the work environment is constructed to make the agent's thinking visible (i.e., an agent must show each of the problem-solving steps rather than just a final solution), so that experts can provide hints and feedback on intermediate steps. If an agent knows what to do next, then it takes action in the work environment. In response, the work environment is updated and annotated with feedback by an expert tutor. The agent receives the expert's feedback from the work environment and uses it to improve its skills. After skill learning, it applies this knowledge towards solving the next step in the modified problem state. In the event that the agent does not know what to do next, it can request a demonstration from the expert who will then provide it directly in the environment. The agent observes this demonstration in the environment and updates its skill knowledge. This process is repeated until the expert is convinced that the agent has successfully learned the target skills. Previous work from the intelligent tutoring system literature has modeled the step-level interactions between an expert tutor and the work environment (Vanlehn, 2006), but not the interactions between the learning agent and the work environment. The Apprentice Learner Architecture (shown in left box of Figure 1) is a computational theory of apprenticeship learning that aligns with the step-level interactions described by VanLehn (shown in the right box of Figure 1).

The Apprentice Learner Architecture posits three learning mechanisms to induce new skills from prior feature and function knowledge and observed demonstrations and feedback. When given a demonstration, the *how* learner uses function knowledge (e.g., a function for adding two numbers) to search for a sequences of functions that can explain the observed demonstration. After discovering a function sequence, the *where* learner acquires general perceptual patterns for recognizing the elements used in the discovered sequence. Finally, the *when* learner uses the tutor state, augmented with feature knowledge (e.g., a feature for recognizing a plus or multiplication sign), to identify the conditions under which the discovered sequence should be executed. The combination of the components discovered by the how, where, and when learners constitute a skill, or a new piece of domain knowledge. After a new skill is learned, it can be applied in subsequent problem solving.

In order to apply learned skills, the Apprentice Learner Architecture posits that learners use a basic Recognize-Act cycle to apply their skill knowledge towards solving problems. When a learner is presented with a problem, they query their skill knowledge to determine if any known skills are applicable. If an applicable skill is found, then it is executed and correctness feedback on the resulting action is utilized by the *when* learner to further refine the conditions under which the skill can be executed. In the event that no skills are applicable, then, as mentioned previously, the learner requests a demonstration that is passed to the how, where, and when learners to produce a new skill.

This architecture has been used to construct models of human apprenticeship learning (i.e., particular choices of of algorithms for the where, when, and how learning) in a fraction arithmetic tutoring system. In a recent study, I showed that generated models were able to predict the outcome of a human problem sequencing experiments in the fractions tutor (MacLellan, Harpstead, Patel, & Koedinger, 2016). Additionally, I demonstrated how the architecture can be used to generate and test alternative models of apprenticeship learning to see which models yield behavior that is more similar to the human behavior.

3. Future Work

I am currently exploring how to better connect the Apprentice Learner Architecture with existing cognitive systems work, such as mapping the different learning mechanisms of my architecture to the learning components of existing cognitive architectures. For example, the where and when learners are similar to the learning mechanism used to refine action conditions in ICARUS and SOAR. Additionally, the how learner utilizes search similar to the problem solvers used by these architectures. However, there are key differences that still need to be addressed, such as how operators and goals should be constructed. Currently, my architecture relies on features and functions that lack domain semantics, so it is unclear how this will impact problem solving in the cognitive architecture. Additionally, my architecture is state, not goal driven. It is unclear how similar behavior can be achieved using a cognitive architecture and where goal structure, which is necessary to direct problem solving in these architectures, would come from.

I am also exploring alternative variations of my architecture that have fewer learning components. For example, the where and when components might be merged because they can both be framed as condition learning and combined. Additionally, it might be possible to unify all three mechanisms into a single mechanisms. I am currently exploring which of these alternative models is better supported by the human tutor data.

Acknowledgements

This research was supported in part by the Department of Education (#R305B090023), National Science Foundation (#SBE-0836012). All opinions expressed in this article are those of the authors and do not necessarily reflect the position of the sponsoring agency.

References

Anderson, J. R. (1993). *Rules of the Mind*. Psychology Press.

Jones, R. M. (1999). Automated Intelligent Pilots for Combat Flight Simulation. (pp. 1–16).

Langley, P., Laird, J. E., & Rogers, S. (2009). Cognitive architectures: Research issues and challenges. *Cognitive Systems Research*.

MacLellan, C. J., Harpstead, E., Patel, R., & Koedinger, K. R. (2016). The Apprentice Learner Architecture: Closing the loop between learning theory and educational data. *The 9th International Conference on Educational Data Mining*.

Vanlehn, K. (2006). The Behavior of Tutoring Systems. *International Journal of Artificial Intelligence in Education*, *16*, 227–265.