Integrating Transfer Learning in Synthetic Student

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

Abstract

Building an intelligent agent, which simulates human-level learning appropriate for learning math, science, or a second language, could potentially benefit both education in understanding human learning, and artificial intelligence in creating human-level intelligence. Recently, we have proposed an efficient approach to acquiring procedural knowledge using transfer learning. However, it operated as a separate module. In this paper, we describe how to integrate this module into a machine-learning agent, SimStudent, that learns procedural knowledge from examples and through problem solving. We illustrate this method in the domain of algebra, after which we consider directions for future research in this area.

Introduction

Building an intelligent agent, which simulates human-level learning appropriate for learning math, science, or a second language, would benefit both the education and AI communities in understanding and creating human-level intelligence. Supporting full range human skill acquisition is one of the essential issues in this task. Programming by demonstration, where systems learn procedural knowledge primarily by observing expert behavior, is similar to one of the major approaches for humans to acquire skills, learning by observation. Recently, we have explored the use of *transfer learning* in programming by demonstration to improve learning efficiency. The learning task we looked at is recognizing key features from an input string in the framework of *Probabilistic Concext Free Grammar (PCFG)*. We have shown promising results in both the algebra domain and synthetic domains.

However, our approach operated as a stand-alone module that was not part of an integrated system. In this paper, we discuss and explore the possiblity of adapting this mechanism into a larger system. In particular, we chose SimStudent (Matsuda *et al.* 2009), a state-of-art machine learning agent that acquires procedural knowledge from examples and through problem-solving experiences. We describe how the learning mechanism can be integrated into SimStudent, and how this integration could reduce the amount of prior knowledge that SimStudent needed in acquiring new knowledge.

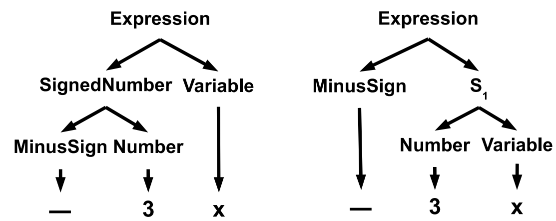

Figure 1: Correct and incorrect parse trees for $-3x$.

Fetaure Recognition as PCFG Induction

A feature recognition task is given a string (e.g. an expression in algebra), to identify a substring that corresponds to some feature (e.g. the coefficient in the expression). One interesting insight is that feature recognition closely connects to PCFG induction. As shown in Figure 1, identifying -3 as the coefficient of $-3x$ relies on correctly parsing the input string to a signed number $-3x$ plus a variable x. Table 1 is an example of the PCFG for expressions of the form $\{Number \quad x\}$ and $\{-Number \quad x\}$ consists of five rules, where $SignedNumber$ in the reduction rule $Expression \rightarrow 1.0,\ [SignedNumber]\ Variable$ is the intermediate symbol corresponding to the key feature "coefficient". Then, the feature recognition task can be formulated as a PCFG learning problem.

We consider two protocols for learning PCFGs. The first one is unlabeled common subtasks. The input of the learning system is just a set of strings. The second is labeled common tasks. The input is a set of string substring pair. Each pair consists of a string, plus its subsequence associated with the key feature. The objective of our learning algorithm is to acquire a PCFG, and identify an intermediate symbol as the key subgoal, so that the acquired PCFG is able to recognize key features from a full string.

A Brief Review of SimStudent

SimStudent is a machine learning agent that acquires production rules from examples and problem solving experience. A production rule consists of a left hand side (LHS) and a right hand side (RHS). LHS specifies a condition as a relation that must be held before firing the rule. RHS includes a sequence of primitive actions that need to be carried out to achieve the goal.

Table 1: Probabilistic context free grammar for coefficient in algebra

Primitive symbols: $-, x$;
Non-primitive symbols: $Expression, SignedNumber$
$Variable, MinusSign, Number$;
$Expression \rightarrow 1.0,\ \ [SignedNumber]\ Variable$
$Variable \rightarrow 1.0,\ \ x$
$SignedNumber \rightarrow 0.5,\ \ MinusSign\ \ Number$
$SignedNumber \rightarrow 0.5,\ \ Number$
$MinusSign \rightarrow 1.0,\ \ -$

Prior to learning, SimStudent is given a library of feature predicates and primitive actions, some of which are generic and some of which are specialized to the target domain (e.g., algebra). To learn from examples, the user needs to demonstrate how to solve problems in a cognitive tutor. For each step during the demonstration, the user first specifies the focus of atention on the interface. He/she then performs a step, and labels the demonstrated step with skill name. SimStudent learns a skill by generalizing and/or specializing the production rule. LHS is acquired by Foil (Quinlan 1990). RHS is learned by searching the shortest operator sequence consistent with all previous examples.

Transfer Learning for PCFG

To build our learning model, we extended an existing PCFG learning algorithm. In particular, we chose the learning mechanism proposed by Li et al. (Li *et al.* 2009), since it acquires PCFG from plain strings without any prior structural knowledge. The system consists of two parts, a greedy structure hypothesizer, which creates non-primitive symbols and associated reduction rules as needed to cover all the training examples, and a Viterbi training step, which iteratively refines the probabilities of the reduction rules. We extended the original algorithm to transfer solutions to common intermediate symbols (features) from one task to another in accelerating the learning process.

We first consider the situation where the user explicitly identifies a shared feature for the learner. The original learning algorithm did not make use of this information. Thus, it sometimes may not be able to learn a grammar that has the feature symbol embedded in it. However, given the user's explicit labeling of one subsequence as fulfilling the common feature, we force the original learner to produce the desired result by learning the subgrammar for the key feature and acquiring the whole grammar separately. However, transfer learning in explicitly labeled common features requires extra work for the users. We considered a second learning protocol, where the shared feature is present, but not explicitly identified. In this case, two tasks share some common features that are not labeled by the user. To make use of previously acquired knowledge, the learner keeps record of the acquired grammar and the number of times each grammar rule appeared in a parse tree. When a new learning task comes in, the learning algorithm first uses the known grammar to build the smallest number of most probable parse trees for the new records, and then switches to the original GSH and acquires new rules based on the partially parsed sequences. During the Viterbi training phase, the learning algorithm adds the applied rule frequency associated with the training problems of the current task to the recorded frequency from previous tasks.

Integrating Transfer Learning in SimStudent

The proposed algorithm only recognizes subsequences from input as features, and thus is unable to generate characters from the input. To build a system that models human-level intelligence, we should extend our system to support not only feature recognition but also problem solving. Recall that SimStudent acquires production rules to solve problems. It requires prior knowledge on both the feature predicates and the primitive actions. The feature recognition process described above belongs to the RHS of the production rule. We could integrate our learning mechanism with SimStudent to enable the power of problem solving. To do this, we can first call the feature recognition learner to acquire procedural knowledge for the RHS, so that the production rule is able to extract important features (i.e. coefficient) from the problem. We then use the original procedure built in SimStudent to learn how to transform the original equation into the next step using the key features generated from the previous steps.

By doing this, since the proposed algorithm does not require any prior knowledge, we have potentially reduced the amount of knowledge SimStudent needs for future learning. Also, since the proposed algorithm is a transfer learning algorithm, it can easily make use of previously acquired knowledge to aid later learning. In this case, SimStudent does not need to remember all previous problems for future learning, which saves both time and space. Last, we believe that the proposed feature recognition system can be naturally extended to a sequence generation mechanism. With that extension, we would be able to fully replace the learning of RHS with the extended algorithm. Then, the learning system will be able to acquire procedural knowledge without given prior knowledge on primitive actions.

Conclusion

In this paper, we motivated the research on learner modeling. We proposed a novel approach that applies transfer learning in programming by demonstration to improve the accuracy and the rate of learning. We demonstrated this approach in the task of recognizing key features from strings. We described how to integrate this learning mechanism into an integrated agent, SimStudent, and explained how this integration could reduce the amount of prior knowledge needed to acquire new knowledge.

References

Nan Li, Subbarao Kambhampati, and Sungwook Yoon. Learning probabilistic hierarchical task networks to capture user preferences. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, Pasadena, CA, 2009.

Noboru Matsuda, Andrew Lee, William W. Cohen, and Kenneth R. Koedinger. A computational model of how learner errors arise from weak prior knowledge. In *Proceedings of Conference of the Cognitive Science Society*, 2009.

J. R. Quinlan. Learning logical definitions from relations. *Mach. Learn.*, 5(3):239–266, 1990.