Collaborative Learning of Hierarchical Task Networks from Demonstration and Instruction

Anahita Mohseni-Kabir, Sonia Chernova and Charles Rich
Worcester Polytechnic Institute, Worcester, MA, USA
{amohsenikabir, soniac, rich}@wpi.edu

Abstract

In this work, we focus on advancing the state of the art in intelligent agents that can learn complex procedural tasks from humans. Our main innovation is to view the interaction between the human and the robot as a mixed-initiative collaboration. Our contribution is to integrate hierarchical task networks and collaborative discourse theory into the learning from demonstration paradigm to enable robots to learn complex tasks in collaboration with the human teacher.

Introduction

Our goal is to advance the state of the art of intelligent agents that can learn complex procedural tasks from humans. Our work draws on several research areas, primarily robot learning from demonstration (LfD) (Argall et al. 2009), hierarchical task network (HTN) planning, and collaborative discourse theory. Our approach is to view the interaction between the human teacher and the learning agent as a mixed-initiative collaboration, in which both parties are committed to the shared goal of successful learning, and in which both parties make contributions in the form of both actions and communication, including verbal instructions, asking questions, and critiquing. For this, we draw heavily on collaborative discourse theory and tools.

Our approach is based on the conjecture that it is often easier for people to generate and discuss examples of how to accomplish tasks than it is to deal directly with task model abstractions. Our approach is thus for the robot to iteratively learn tasks from human demonstrations and instructions. In addition, the robot is able to ask questions, to which human responds. By engaging the robot as an active partner in the learning process, and by using the hierarchical structures, we believe that complex tasks can be naturally taught by non-expert users. Our work makes contributions in the following areas:

1. a unified system that integrates hierarchical task networks and collaborative discourse theory into the learning from demonstration;
2. a novel approach for learning task structure from a small number of demonstrations, including the task hierarchy, temporal constraints and inputs/outputs of a task;
3. novel generalization techniques that reduce the number of demonstrations required to learn the task through input generalization and merging;
4. integration of mixed-initiative interaction into the learning process through question asking.

System Architecture

Figure 1 presents an overview of our system architecture; the key collaborative learning components are highlighted in blue. The architecture consists of two subsystems, learning and execution.

The inputs to the system are user demonstrations, instructions and answers to questions provided via a GUI. The core components of the learning subsystem are the Task Structure Learning, Generalization and Question Asking modules, which together generate an HTN model of the task being learned. The entire learning process is supported by Disco, an implementation of collaborative discourse theory (Grosz and Sidner 1988; Rich, Sidner, and Lesh 2001) and ANSI/CEA-2018 (Rich 2009). Disco's dialog management capabilities are used to maintain a focus stack which keeps track of the current topic and has expectations for what needs to be said or done next. During the execution of a learned task, the Disco planner decomposes each non-primitive task in the HTN into its subtasks. When the planner reaches a primitive task, the primitive task is sent as an action command to the execution subsystem. The execution subsystem includes the Motion Planner, Abstract World Model (AWM), and the learning agent embodied either as a physical PR2 or as a Gazebo simulation.

Learning Modules

In this section we briefly summarize the key learning components of the system.

Generating the Task Hierarchy: One of the core functionalities of the learning subsystem is to learn the implicit hierarchical structure of the demonstration sequence. Each task in an HTN has one or more recipes, or methods for decomposing non-primitive actions. Each recipe specifies a set

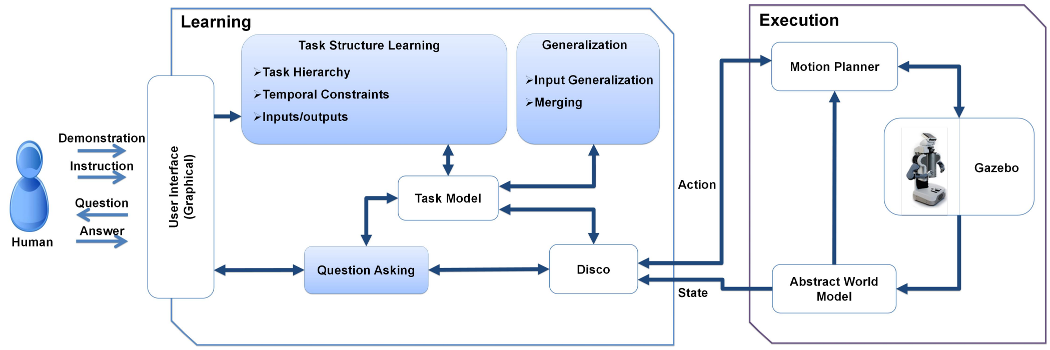

Figure 1: An overview of the system architecture.

of steps that are performed to achieve the non-primitive action that is the collective objective of the steps (e.g., rotate tires in an x-pattern or front to rear). In each new demonstration sequence, the user either teaches a new non-primitive action (e.g., unscrewHub) or demonstrates a new recipe for an existing nonprimitive action (e.g., xPattern) by using existing non-primitive and primitive actions. The system gives the flexiblity to the human teacher to switch between bottom-up approach and top-down approach at any point in the teaching process of the task model.

Associating Inputs/Outputs: Each task within the HTN has zero or more *inputs* and *outputs* associated with it. The input of a task must be specified by the user at demonstration time, thus allowing the input to be added to the task definition. The system is also able to determine the output of a task, and analyze the dependency between actions by propagating this information throughout the hierarchy.

Learning the Temporal Constraints: Demonstrations are intrinsically totally ordered, i.e., any set of (discrete, non-overlapping) actions performed in the real world occur in sequence. However, in many cases, only some of the demonstrated ordering is fixed. For example, in tire rotation task, even though you must demonstrate unhanging all four tires in some order, the order does not matter. Learning the minimum required ordering constraints is important to have a more flexible and reusable plan. One of the contributions of the paper is an automated algorithm for finding the temporal constraints between steps in a recipe. Past approaches have mostly focused on learning these constraints from multiple demonstrations, which requires many demonstrations in a task with many steps. In addition to using past approaches, we are using a new technique (Mohseni-Kabir, Rich, and Chernova 2014) based on finding the data flow betweeen steps, which enables us to learn these temporal constraints from a single demonstration.

Generalization: The generalization module performs two functions, input generalization and merging. Generalizing over the inputs of a task is useful because it makes the task structure more reusable and flexible, and because it reduces the number of demonstrations required for learning the task. Merging is applied when the human provides multiple demonstrations of the same task, in which case the system merges them to allow for generalization across the two examples. This approach has three advantages: 1) this avoids adding a separate recipe for each tiny difference between the new demonstration for a specific task and the previously learned model, 2) by merging different demonstrations and factoring the common steps between them, we are postponing recipe selection until the choice must be made during execution, resulting in a more robust system, and 3) it reduces the number of demonstrations required to learn the task.

Question Asking: Since the robot and teacher represent knowledge differently, the teacher does not always know what additional information the robot requires. The robot is able to expedite the learning process by asking questions when it lacks information, thereby further reducing the number of demonstrations required to learn the task and helping the human teracher to build the hierarchy. Our current system supports six different question types.

Evaluation

We evaluated our complete system in a preliminary pilot study using the Gazebo simulation of a car maintenance domain. Specifically, the robot was taught a tire rotation task which consists of first removing the tires by uncrewing and unhanging the hubs, then rotating the tires in one of two patterns (i.e., two alternative recipes), hanging the tires and then screwing them on. Tire rotation was chosen because the task is relatively simple, requiring only six unique primitive actions (PickUp, PutDown, Hang, Unhang, Screw, Unscrew), but highlights the benefits of using the HTN representation, including alternative recipes, hierarchy and inputs/outputs. The results of this study shows that using the methods described in this paper, we are able to teach the complete task structure, including alternative recipes, in 26 demonstration steps. Importantly, we note that the complete execution of the tire rotation task (both recipes) requires 128 steps. Thus, remarkably, we are able to teach not only one, but two, ways of performing this task (i.e., both variants of Rotate) in fewer steps than it takes to perform the task once all the way through.

Acknowledgments

This work is supported in part by ONR contract N00014-13-1-0735.

References

Argall, B. D.; Chernova, S.; Veloso, M.; and Browning, B. 2009. A survey of robot learning from demonstration. *Robotics and Autonomous Systems* 57(5):469–483.

Grosz, B. J., and Sidner, C. L. 1988. Plans for discourse. Technical report, DTIC Document.

Mohseni-Kabir, A.; Rich, C.; and Chernova, S. 2014. Learning partial ordering constraints from a single demonstration. In *ACM/IEEE International Conference on Human-robot interaction*, 248–249.

Rich, C.; Sidner, C. L.; and Lesh, N. 2001. Collagen: applying collaborative discourse theory to human-computer interaction. *AI Magazine* 22(4):15.

Rich, C. 2009. Building task-based user interfaces with ANSI/CEA-2018. *IEEE Computer* 42(8):20–27.