Training wheels for the robot: Learning from demonstration using simulation

Nathan Koenig
Open Source Robotics Foundation
Mountain View, CA

Maja Matarić
University of Southern California
Los Angeles, CA

Introduction
Learning from demonstration (LfD) is a promising technique for instructing/teaching autonomous systems based on demonstrations from people who may have little to no experience with robots (Billard et al. 2008). A key aspect of LfD is the communication method used to transfer knowledge from an instructor to a robot. The communication method affects the complexity of the demonstration process for the instructor, the range of tasks the robot can learn, and the learning algorithm itself.

We have designed a graphical interface and an instructional language to provide an intuitive LfD teaching system. The challenge of simplifying the teaching interface is that the resulting demonstration data are less structured, adding complexity to the learning process. In our approach, this additional complexity is handled through the combination of using a minimal set of predefined behaviors and a task representation capable of learning probabilistic policies over a set of behaviors. The predefined behaviors consist of finite actions a robot can perform, such as move to, pick up, and put down. Behaviors act as building blocks for more complex tasks, and are parametrized by features, objects the robot can observe and manipulate.

A series of behaviors and features from a group of instructors is used to produce a generalized policy for a task. The policy is represented by a set of influence diagrams, an extension of Bayesian networks, that incorporate the ability to probabilistically choose actions based on state information. We allow for error in the teaching and learning process by providing a mechanism to refine influence diagrams during autonomous operation. This technique effectively reduces the number of complete demonstrations required for a robot to accurately learn a task.

Teaching Interface and Task Learning
Teaching interfaces use a variety of forms, including manual manipulation (Hersch et al. 2008), teleoperation via joysticks (Grollman and Jenkins 2007), graphical interfaces (Chernova and Veloso 2008), external observations (Schaal, Ijspeert, and Billard 2004), and sensors placed on the instructor (Ijspeert, Nakanishi, and Schaal 2002).

These methods necessarily trade off teaching complexity and complexity of the learning algorithm.

Our approach emphasizes ease of use for instructors and generality of use with multiple robot platforms. The reconfigurable graphical interface meets these needs by providing an experience that is familiar to people with some computer experience. The instructor uses the graphical interface to observe the state of the world and send instructions to the robot, as shown in Figure 1.

The instructor commands the robot by building sentences using an instructional language that is composed of a behavior, a feature, and an optional modifier-feature combination that specifies how the behavior is performed relative to the feature. An example instruction is put down the salt on the table, where put down is the behavior, salt is the feature, and on the table is the optional modifier-feature combination.

The graphical interface guides the instructor through the process of constructing a sentence by sequentially asking for the next part of the instruction. An undo option allows the user to easily fix mistakes. Once the instruction is complete, it is sent to the robot for execution. During execution, the user observes the robot’s progress and any textual feedback from the robot. The textual feedback consists of sentences that describe the robot’s internal state and response to instructions. Feedback examples include, “Moving to saucepan”, “Unable to reach bowl”, or “Waiting 5 seconds”.

The process of building instructions continues until the instructor decides the task demonstration is complete.

During the demonstration, the robot records the instructions and world state information. After one or more instructors have provided demonstrations, the recorded information is used to learn a task policy. The task is represented by a set of influence diagrams constrained to non-realtime and discrete sequences of behaviors. The influence diagram learning algorithm and example influence diagrams are detailed in (Koenig, Takayama, and Matarić 2010).

Experimental Setup
The graphical interface and learning system were used to teach a robot the process of preparing mushroom risotto. The complete set of predefined behaviors consisted of move to, pick up, put down, stir, chop, turn on, turn off, and wait, which represent the minimum set needed to complete the risotto task.
that collected demographic information. During the demonstration, each participant completed a survey immediately after the task completion. At this point, each participant was free to instruct the robot. Following the demonstration, each participant completed a survey that collected demographic information.


The demonstration environment used the Gazebo (Koenig and Howard 2004) simulation environment, with dynamics disabled for improved speed. The environment consisted of a kitchen, a PR2 robot, ingredients, and utensils. Simulation was used to reduce teaching time, provide a stable environment, and facilitate error correction through an undo feature that reverses instructions.

A fifteen-step tutorial guided participants through the graphical interface. Upon completion of the tutorial, participants received a printed mushroom risotto recipe. At this point, each participant was free to instruct the robot. Following the demonstration, each participant completed a survey that collected demographic information.

Results

Thirty three participants completed the demonstration, 26 male and 7 female, with an age range of 23 to 67. The average time to complete the demonstration was 27 minutes, with a standard deviation of 10 minutes. In contrast, a PR2 robot took roughly three to four times as long to complete the task due to the pace of perception and slowness of the movements for safety.

As expected, the instructions sent to the robot were similar across participants in the beginning of the demonstration. The instructions diverged over time as participants chose to complete the recipe using a different order of instructions. This divergence made it more difficult to learn correct influence diagrams. As a result, a few influence diagrams produced incorrect behavior when used on an autonomous robot.

The incorrect diagrams can be discovered by the system itself or by a human observer. An incorrect influence diagram can be self-discovered when multiple behaviors have the same probability. In these cases, the system is incapable of choosing the best behavior and must ask for help. In all other instances, the system chooses what it believes to be the best behavior, which may not in fact be what a human observer would select.

In both cases, a human may intervene and provide the system with the correct behavior. This new information is incorporated directly into the influence diagram through an update process.

Discussion and Future Work

We have developed an approach that provides an intuitive interface for instructing a robot in time-extended tasks. The interface requires no prior knowledge of robotics. Teaching in simulation provides a less time consuming teaching experience with the ability to gracefully fix errors. The demonstration data are used to generate influence diagrams that represent the task being taught. These diagrams can be refined when the system operates autonomously.

Due to the influence diagram representation, we are constrained to high-level tasks that are categorized as a time-extended series of behaviors. The process of solving influence diagrams does not operate on the time scale necessary for robot joint angle control. We are also limited by the range of features and behaviors available to the robot. The diversity of features is ever increasing, but is still limited by current work in object perception and recognition.

Our future work will explore incorporating knowledge transfer across tasks to allow the robot to utilize influence diagrams from one task to reduce teaching time for a new task. We will also study the use of graphical interfaces as a teaching tool, and determine how best to provide two-way communication between an instructor and a robot.

References


