Interactive Task Learning with Discrete and Continuous Features

Crystal Chao, Maya Cakmak, and Andrea L. Thomaz *
Center for Robotics and Intelligent Machines
Georgia Institute of Technology, Atlanta, GA
[cchao, maya, athomaz]@cc.gatech.edu

Abstract

Learning tasks from demonstration is key to the flexibility of robots and their accessibility to non-programmers. We present a task learning framework that combines the strengths of discrete and continuous representations. The robot learns a set of *criteria* and *expectations* to represent the goal of a demonstrated task. The task consists of performing actions that fulfill expectations on objects that meet the criteria. We propose modeling continuous criteria and expectations with Gaussian distributions. To deal with simultaneous demonstration of multiple tasks, we assume that expectations can be multi-modal and model them as mixtures of Gaussians. We present an implementation of this framework on the robot Jimmy.

Introduction

The problem of learning by demonstration for a robot or autonomous agent is the acquisition of knowledge or behavior from examples demonstrated by a human. Robots that learn from demonstration have the potential to offer more flexibility and accessibility to non-programmers. Learning by demonstration can also provide a good initialization or heuristic for the robot's learning, especially for learning problems with large or even infinite search spaces.

Learning by demonstration has been approached from many different directions, including motion learning for motor control (Calinon and Billard 2008), action policy learning (Chernova and Veloso 2007), and graph-based planning (Zoliner et al. 2005; Nicolescu and Mataric 2003). These systems select either a continuous or discrete representation for learning. Discrete and continuous models both have their uses and tradeoffs. The high-level structure offered by discrete models is amenable to reasoning using traditional artificial intelligence, such as well-established partial-order planning algorithms or first-order logic systems. On the other hand, continuous models offer the benefit of more granularity when representing the data, as well as reasoning using confidence or uncertainty measures.

With appropriate perception, one can determine hierarchical labels for objects (such as "tool" being a parent of "screwdriver") and learn which objects are relevant to a particular action or task. It becomes more ambiguous how to reason about world states when using continuous-valued data. Ideally one wants to capture the structure of the continuous space using a few parameters in a way that can still be usable by a generative action model. Our goal in this study is to reason with both types of data within a single framework.

In previous work, Bayesian maximum-likelihood learning was used for learning task goals from demonstrations (Breazeal and Thomaz 2008). In this framework, the goal is not provided a priori to the robot; instead, the robot considers all hypotheses consistent with the demonstrations and learns the maximally specific hypothesis as the goal. Hypotheses are enumerated on a version space over the discrete features. In this study, we extend the described task learning method by (1) proposing a method that uses both discrete and continuous features, and (2) presenting a way to learn multiple subtasks simultaneously from a single demonstration.

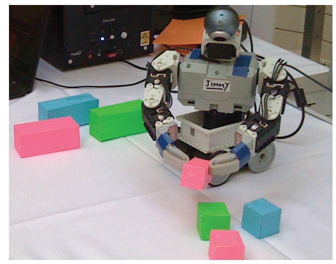

Figure 1: Robot Jimmy executing a sorting task.

Approach

Our task learner expands upon the Bayesian learner described in (Breazeal and Thomaz 2008), which reasons about tasks in the following manner. The robot takes a snapshot of the perceived state of the world preceding and following every demonstration of a task. Perceptual features (attributes) that remain unchanged are labeled as *criteria*, and perceptual features that change are labeled as *expectations*. Task hypotheses are constructed as combinations of criteria and expectations over a version space. The best hy-

*This work is supported by the National Science Foundation.

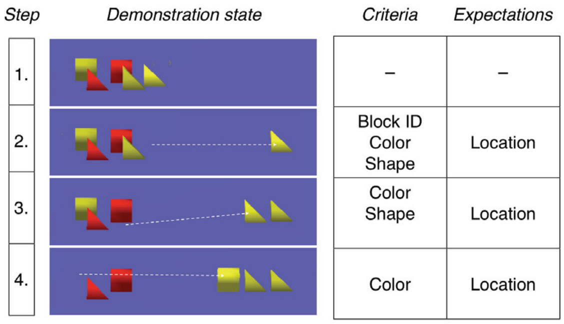

Figure 2: Sample task demonstration and the criteria and expectations extracted after each step of the demonstration.

pothesis is the one of appropriate generality or specificity that maximizes the coverage count over the demonstrated examples. Executing the task consists of performing actions that fulfill expectations on objects that meet the criteria.

Adding continuous features

In order to extend this method for continuous features, we calculate the mean and variance of continuous-valued features, treating each continuous feature independently. This gives the robot an understanding of what it means to be a representative example in this feature space, as well as how much freedom it has when reproducing the task goal.

Criteria. The mean and variance are calculated for continuous features that do not change throughout the demonstrations. Novel objects are compared against this model to assess whether they meet the criteria of the learned task. For example, the "height" of books that go on a particular shelf is a continuous attribute that does not change during a demonstration and has some variance over different demonstrations given different books. One can decide whether a new book of a different height would go onto the same shelf according to the distribution of the heights of demonstrated books.

Expectations. Similarly, the mean and variance are calculated for continuous features that do change during the demonstration. This model tells the robot what is expected for objects that meet the task criteria. The mean represents the ideal expected state of the object, and the variance represents the allowable deviation from the ideal. For example, "location" and "orientation" of books are continuous attributes that change during the demonstration of putting books on a bookshelf. While the locations of the books may have a high variance over the length of the bookshelf, the orientations may have a small variance around a vertical position.

Extracting subtasks

The second issue addressed in this study is the simultaneous demonstration of multiple subtasks. The previous version of the learner required that each subtask be demonstrated and labeled separately. However, it is often more convenient to demonstrate multiple tasks simultaneously. In a household sorting scenario, the previous version would require the teacher to first demonstrate that all the books go on one shelf and label it as one task, then separately demonstrate that all trash goes into the trash bin and label it as a separate task. If both tasks were demonstrated together, they would be taken as one collective task. The resulting criteria would cover both books and trash, with either being destined for the bookshelf or the trash bin. To deal with the issue of automatically differentiating subtasks, we add the assumption that the expectations of tasks can be *multi-modal*.

Discrete features. If a discrete expectation has more than one possible end value, a separate subtask is created for each end value. For example, if a set of white colored objects are being painted to red and blue, "coloring red" and "coloring blue" are considered two different tasks, and different criteria are formed for each task. This potentially allows the learner to capture different requirements, such as size or shape of the object, for being colored blue or being colored red.

Continuous features. For continuous features, having multi-modal criteria means that instead of fitting a single Gaussian over the changed continuous features, we fit a mixture of Gaussians. Each Gaussian in the mixture corresponds to a separate subtask. In this way the different locations for sorting different objects can be learned from a demonstration of sorting different types of objects simultaneously.

Implementation

In our demonstration, a robot learns the goal configuration of a set of simple blocks by observing a person's demonstrations. The robot then executes the task individually or cooperatively with the person.

Robot Platform. We use the robot Jimmy, which is an upper-torso humanoid on wheels built from Bioloid kits and a webcam (Fig. 1). Its 8 degrees of freedom enable arm movements, torso rotation, and neck tilt. The wheels are used to navigate the workspace. The webcam's head-like appearance functions as a transparency mechanism for the human to identify the robot's object of attention during the interaction.

Perception. Tasks demonstrated to Jimmy consist of changing the state of objects in its environment. The objects are perceived through a fixed overhead camera that captures Jimmy's entire workspace. Images are filtered for a set of pre-defined colors, and multi-blob tracking is performed on each filtered image. Then a set of features is extracted from each blob. The *location* of the object corresponding to the blob is determined using a homography from the table plane to the camera image plane. The objects are also represented by two discrete features: *color* and *shape*. There are three possible colors (pink, green, and blue) and two possible shapes (cube or long prism). Shape is determined by thresholding the ratio of the blob's major axis to the blob's minor axis.

The robot detects people by skin color being present in the workspace, as well by perceptual state changes that occur independently of the robot's own actions. The starts and ends of demonstrations are demarcated using a graphical user interface or speech commands.

Behaviors. Jimmy's behaviors are controlled by a state machine that has three main states: (i) observing a demonstration, (ii) individual execution, and (iii) cooperative execution. The robot enters the demonstration state when either (i) all the objects in the workspace are removed, or (ii) a demonstration is triggered using the GUI or a speech command. When a demonstration starts, the robot initially gazes towards the person, then gazes at the objects being manipulated. When the demonstration ends, the robot models the goal of the demonstrated task and transitions into execution mode.

During execution, the robot tries to preserve the goal of the learned task. If an object is perturbed from the goal state, the robot attempts to restore the goal configuration. If the object is within reach of the robot, the robot will (i) navigate to the object, (ii) pick up the object, (iii) navigate to the goal location, and (iv) place the object. Similarly, if a novel object is introduced and placed in the workspace such that it does not satisfy the current goal, the robot will pick up the object and place it in the correct location. The robot uses open-loop navigation based on prior calibration of duration of travel with a fixed velocity. When all the objects in the workspace are at their goal locations, the robot just looks around.

If a person is present during execution, the robot solicits the person's help by (i) gazing towards the person, (ii) pointing to an object that needs to be moved, and (iii) pointing to the goal location for that object. During cooperative execution, the robot gives priority to objects that are out of its reach in order to benefit maximally from the person's help. If the action indicated by the robot is completed by the person, the robot nods to confirm the action and moves on to the next object. If the person does not respond to the robot after pointing to the object three times, the robot points to the object and raises its arms asking for the object. If the person hands the objects to the robot, the robot takes the object and places it at the goal location. If the person does not reply to three such queries, the robot gives up and starts individual execution.

Learning Task. The learning task in this study is simplified by some assumptions and constraints. The number and type of features of objects used in the tasks is limited. As mentioned previously, the task space includes one continuous feature (location) and two discrete features (color and shape). The location feature is further simplified by projecting the 2-D position of the objects onto Jimmy's 1-D

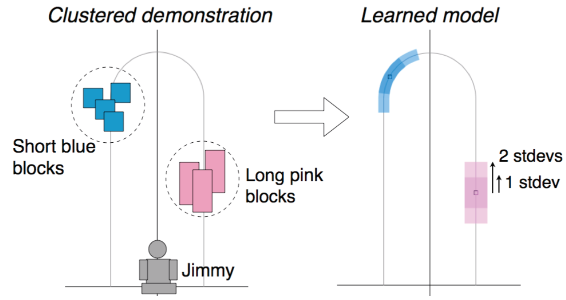

Figure 3: Jimmy's workspace.

workspace, the region that the robot can reach by translating along a straight line and rotating its torso (Fig. 3).

The fact that some features can be changed while others cannot also limits the number of different tasks that can be demonstrated. In this case, location of objects can be changed while shape and color cannot. Therefore, the possible criteria are color and shape, and the only possible expectation is location. With 1-D location being the only possible expectation, we cluster the locations based on a fixed distance threshold and fit a single Gaussian on each cluster rather than fitting a mixture of Gaussians (Fig. 3).

For this particular demo, we do not provide an interface to demonstrate and label more than one task. Every time a demonstration for a new task is given, the previous task is erased. The start of a demonstration is always assumed to be an empty workspace. Even if only part of the workspace was changed during the demonstration, all objects in the workspace are considered to be part of the task goal.

Results

The described system was demonstrated at the robot exhibition in IJCAI 2009, Pasadena. The robot was able to learn different tasks demonstrated by visitors and perform the correct actions to reconstruct the goal.

Sample Demonstrations. Fig. 4 shows a few examples of demonstrations given during the exhibition, together with tests provided by the demonstrators to assess what the robot learned and the robot's responses to the test actions.

The demonstrations described in Fig. 4(a), 4(b) and 4(c) involve two maximally specific clusters with both color and shape criteria. In this case, only objects that are identical to the ones used in the demonstration meet the criteria for the learned subtasks and are placed next to the other objects (Fig. 4(a)). Other objects are placed outside the workspace. The demonstration given in Fig. 4(d) involves some variation within clusters (*i.e.* shape), so each subtask only has one criterion feature (color). Similarly, in Fig. 4(e) and 4(f), color varies within clusters and the subtasks have one criterion feature (shape). After the demonstration, when the robot is presented with an object it has not seen before (pink cube or long green prism), it is able to place it in the right cluster according to shape. Fig. 1 shows a picture of Jimmy

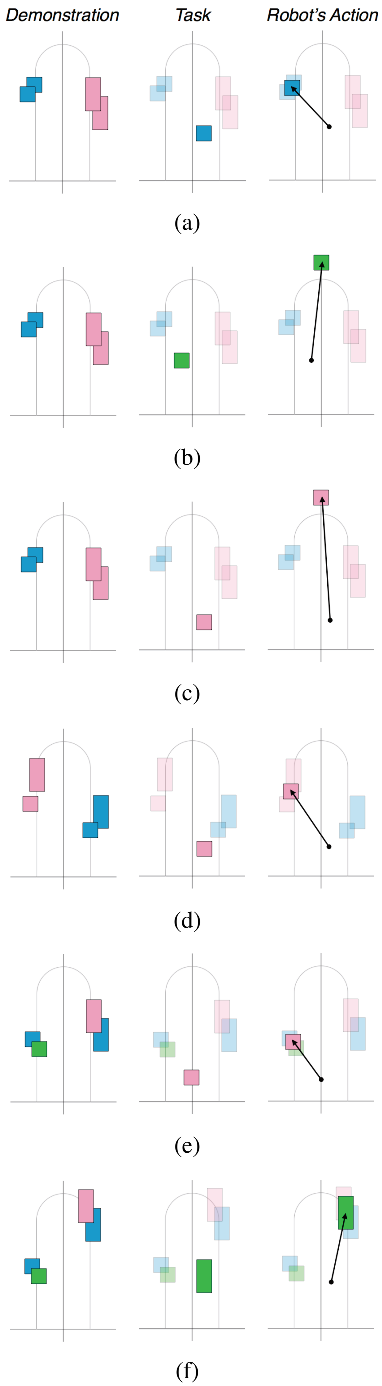

Figure 4: Sample demonstrations and tasks.

taken during the exhibition while executing a task similar to the one described in Fig. 4(e).

Issues. Most issues encountered during the exhibition were related to noise in vision due to changing lighting conditions or errors during open-loop navigation. Slippage in the wheels caused the home position of the robot to drift over time. As a result, the robot sometimes failed to pick up objects and continued executing as if it had the object. When this occurs, the robot realizes the failure only after it has gone back to its home position. This can be addressed using feedback from the webcam on the robot to make sure the object has been picked before proceeding to place it.

Another issue was the discrepancy between people's representation of the task and the robot's actual model. As a result, the robot did always behave as they expected. For instance, in the task given in Fig. 4(b), some subjects reported that they expected the robot to place the green cube with the blue cubes because it shared the shape feature with this cluster but not with the other. They did not expect the maximally specific consistent hypothesis, which designated the color blue as a criterion. People's assumptions about the task space are important to consider in the learning by demonstration problem.

Conclusion

We present two extensions of a task learning framework developed in previous work. Tasks are represented as collections of *criteria* and *expectations*, where the task consists of performing actions that fulfill expectations on objects that meet the criteria. We extend this framework (i) to handle continuous features by representing them with Gaussian distributions, and (ii) to extract subtasks automatically (as opposed to relying on user labeling) by assuming that expectations can be multi-modal. We demonstrate the extended learning framework with simple block sorting tasks on the robot Jimmy.

References

Breazeal, C., and Thomaz, A. L. 2008. Learning from human teachers with socially guided exploration. In *Proc. of IEEE ICRA*.

Calinon, S., and Billard, A. 2008. A probabilistic programming by demonstration framework handling skill constraints in joint space and task space. In *Proc. of IEEE/RSJ IROS*.

Chernova, S., and Veloso, M. 2007. Confidence-based policy learning from demonstration using gaussian mixture models. In *Proc. of Intl. Conf. AAMAS*.

Nicolescu, M. N., and Mataric, M. J. 2003. Natural methods for robot task learning: Instructive demonstrations, generalization and practice. In *Proc. of Intl. Conf. AAMAS*.

Zoliner, R.; Pardowitz, M.; Knoop, S.; and Dillmann, R. 2005. Towards cognitive robots: Building hierarchical task representations of manipulations from human demonstration. In *Proc. of IEEE ICRA*.