Representing Skill Demonstrations for Adaptation and Transfer

Tesca Fitzgerald, Ashok K. Goel, Andrea Thomaz
School of Interactive Computing
Georgia Institute of Technology, Atlanta, GA 30332, USA
{tesca.fitzgerald, goel, athomaz}@cc.gatech.edu

Abstract
We address two domains of skill transfer problems encountered by an autonomous robot: within-domain adaptation and cross-domain transfer. Our aim is to provide skill representations which enable transfer in each problem classification. As such, we explore two approaches to skill representation which address each problem classification separately. The first representation, based on mimicking, encodes the full demonstration and is well suited for within-domain adaptation. The second representation is based on imitation and serves to encode a set of key points along the trajectory, which represent the goal points most relevant to the successful completion of the skill. This representation enables both within-domain and cross-domain transfer. A planner is then applied to these constraints, generating a domain-specific trajectory which addresses the transfer task.

Introduction
Skill transfer is necessary for autonomous robots to learn quickly and adapt to new situations. Without the ability to transfer previously learned skills to a newly encountered problem, a robot would require additional time to learn the new skill from scratch. Leveraging previously learned skills to bootstrap the learning of new skills would significantly reduce the amount of time necessary to learn a new skill, and would require less effort of the skill teacher.

Additionally, skill transfer enables a robot to handle unfamiliar problems with flexibility. When encountering an unfamiliar problem, the robot can attempt to transfer a previously learned skill to address the new problem. We refer to a “target problem” as a new problem for which a skill demonstration has not been provided. It is difficult, however, to define what qualifies as a “transfer” problem. We refer to two classifications of skill transfer problems (Thagard 2005; Goel 1997):

• Within-domain adaptation occurs when the robot transfers a known skill to a target problem which requires the same goal and involves similar objects as the learned skill. The known skill’s action model must be modified to address the target problem. An example of a within-domain adaptation problem is a book-shelving task in which the target problem involves a book with different dimensions than that of the training object. The known skill’s action model can be used to address the new problem with only minor parameter adjustments.

• Cross-domain transfer involves transferring an action model to a target problem in which the same goal must be attained, but a different set of features are used. For example, stacking a set of plates and stacking cups are two tasks which require the same goal, but are completed by referencing different sets of features. Transferring the plate-stacking skill to the target problem may require more than a simple parameter adjustment, and may even require a different trajectory to achieve the new goal.

We focus on interactive skill transfer, where the robot is taught new skills via interactive demonstrations. We define two skill representations that dictate what aspects of a skill demonstration are stored and made available for transfer processes. Our purpose in this work is to consider what representations may enable a robot learner to go beyond simple mimicking (skill adaptation) and get closer to true imitation (transfer).

Background
Learning from Demonstration
Learning from Demonstration (LfD) is an approach to skill learning in which a skill is taught via interactive demonstrations provided by a human teacher (Argall et al. 2009). As the teacher does not directly program the robot, this mode of skill learning is particularly suited for end-users without robotics experience. Skill demonstrations can be provided via several forms of interaction including kinesthetic teaching and teleoperation (Argall et al. 2009). We focus on kinesthetic teaching, in which the teacher physically moves the robot’s arm to demonstrate the skill (Akgun et al. 2012a).

There are several ways of providing a kinesthetic demonstration. One method is to provide a single, uninterrupted demonstration, during which the robot records its joint state at regular intervals throughout the entire trajectory. In another method, the robot collects joint states only in sparse intervals specified by the teacher (Akgun et al. 2012b). Instead of recording the entire motion, the robot records only...
**Social Learning**

We approach skill adaptation and transfer using skill representations inspired by the social learning paradigm. Tomasello describes several ways in which a learner can respond to and represent interactive skill demonstrations (Tomasello 2001):

- **Stimulus enhancement** occurs when the learner is encouraged to interact with an object after observing the teacher drawing attention to the same object. This enables the learner to focus on objects that the teacher determines to be important or useful.

- **Emulation learning** involves a learner interacting with an object after viewing the teacher demonstrate the object affordances. The learner copies the teacher’s actions to explore new ways of interacting with the object.

- **Mimicking** occurs when the learner pays attention to the actions performed by the teacher, without regard for the goal of the teacher’s actions.

- **Imitative learning** involves the learner paying attention to the goals or intention behind the teacher’s actions.

We focus on building skill representations inspired by imitative learning. One interpretation of the difference between mimicking and imitation is the target of the learner’s attention (Cakmak et al. 2010). By this interpretation, learning from imitation occurs when the learner represents an observation as the intention behind that action (Tomasello 2001). Rather than directly copying the demonstrated actions or method used to achieve the goal, the learner may choose to use an alternative method that achieves the same goal. In a robotics context, this may involve a robot learner using a different trajectory than what was demonstrated, provided that the demonstrated goal is still achieved by transferring the observed constraints (Fitzgerald and Goel 2014). A skill learned by imitation is likely to be represented computationally as a goal or set of sub-goals that are integral to the skill. It would be useful to represent the demonstrated skill as a goal-subgoal tree to learn by imitation.

In contrast, learning from mimicking involves replicating the teacher’s actions with emphasis on the method of completing the skill, rather than only learning the goal of the skill (Tomasello 2001). A robot that mimics, rather than imitates, would choose to use a trajectory that is derived from the demonstrated trajectory. Much of the existing work in skill learning by demonstration would qualify as mimicking in that the primary purpose of the learned model is to faithfully reproduce the motion that was demonstrated. We focus on defining skill representations which enable a robot learner to perform both simple mimicking (skill adaptation) and true imitation (transfer).

**Approach**

We approach the tasks of within-domain skill adaptation and cross-domain skill transfer by using representations guided by mimicking and imitation. Action models learned by imitation and mimicking may enable different forms of transfer. We discuss two approaches to representing skills learned via kinesthetic demonstrations such that they can then be transferred.

**Trajectory-Based Representation**

We approach mimicking computationally as a skill representation that encodes both the demonstrated trajectory and the achieved goal. Many approaches exist for representing recorded trajectories, such as Gaussian Mixture Models (GMMs) and Dynamic Motion Primitives (DMPs) (Schaal 2006). We chose to investigate the transferability of skills represented as DMPs, sub-skills that can be adjusted parametrically. They are used to provide flexibility in complex skills; rather than learn the entire skill as a single unit, the skill can be broken down into sub-skills, or motion primitives, that can be learned and parameterized separately, or chained to form a more complex skill (Pastor et al. 2009; Niekum et al. 2012).

We applied DMPs to learning a task consisting of sliding an object (the yellow block shown in Figure 1) and then placing it on a target (near the red block and on the green square). This task was represented by two sub-skills, sliding and picking/placing, which were each represented as a DMP. The DMP policy holds that \( u = \pi(x, \alpha, t) \), where the control vector \( u \) is dependent on the continuous state vector \( x \), time \( t \), and the parameter vector \( \alpha \) (Schaal 2006), which we define as consisting of the following data:

- The robot’s initial end-effector position (as cartesian coordinates) and rotation (as quaternions), represented by \( < x, y, z, qx, qy, qz, qw > \)

- The position of the block, represented as the tuple \( < x_f, y_f > \) containing the offset, in cartesian coordinates, of the center of the block’s left edge, provided by the image capture data.

- The intended goal location of the block, represented as the tuple \( < x_i, y_i > \), containing the offset, in cartesian coordinates, of the right edge of the block, provided by the image capture data.

**Preliminary Results**

By splitting the task into two sub-skills, each represented by a DMP, the task could be transferred to modified examples of the same skill, such as when the objects were slightly offset from the original demonstration. The object offset can be determined using the robot’s depth sensor or RGB camera data. This method is best suited for skill adaptation problems in which the two situations, the
skill to be transferred and the target, are similar in both the
types of the objects used and the method of achieving the
goal. However, this skill representation may not allow for
cross-domain transfer, such as transfer to scenarios in which
the demonstrated trajectory cannot be used.

This approach enables mimicry, as both the goal and
method in which the goal was achieved (the trajectory
recorded during the demonstration) are encoded in the skill
representation. This illustrates how mimicry-based
approaches to skill learning enable within-domain transfer.

**Constraint-Based Representation**

We propose a second skill representation based on imitation
learning, in which *only the goal* of the demonstration
is encoded, rather than the trajectory used to demonstrate
the skill. Our approach builds from the interactive learning
methods described previously.

This representation encodes a skill as a set of goals, which
each consist of a set of constraints. Two types of constraints
are essential to a skill representation: the geometric con-
straints, such as the location of the robot’s end-effector at
the start and end of the skill demonstration, and the dynamic
constraints, such as the end-effector velocity at key points of
the skill demonstration. Rather than encode the full tra-
djectory, this method encodes only what is essential to the skill
by building a set of constraints. These constraints are then
used to plan a trajectory such that the skill is successfully
imitated.

In contrast to the trajectory-based method, which required
a single trajectory demonstration, this method requires both
a keyframe demonstration and a trajectory demonstration.
The keyframe demonstration serves to specify the geometric
constraints of the skill, such as the initial and final configu-
rations of the robot’s arm in 7-DOF joint space. The begin-
ning and end states of the skill will always be represented
as keyframes, with additional keyframes potentially indi-
cated at other points throughout the skill demonstration.
The keyframes are then represented as a set $G = \{g_1, ..., g_n\}$,
where $n > 1$ and each element $g_i$ represents a geometric
constraint of the skill. Each geometric constraint is encoded
as the 7-element tuple containing the joint positions of the
robot’s arm. The speed of the robot’s end-effector is not en-
coded in these keyframes.

Once the geometric constraints are provided via a
keyframe demonstration, the next step is to provide the dy-
amic constraints. A trajectory demonstration of the skill
is given at full speed, stopping only when the goal of the
skill has been reached. The number of robot pose record-

ings, $n$, is specified as $n = t_f / t$, where $t_f$ is the duration of
the entire trajectory demonstration and $t$ is the time interval
between robot pose recordings. The trajectory can be re-
presented by the set $D = \{d_1, ..., d_n\}$ where $d_i$ is the robot
pose at time interval $i$.

The set of geometric and dynamic constraints are then
aligned, creating a set of constraints such that each geo-
metric constraint is associated with a velocity constraint.
A nearest-neighbor approach is used to pair each keyframe
$g_i \in G$ to the pose $d_j \in D$ with the closest end-effector
position, where $d_j$ occurs within a certain time period after

![Image](a) Demonstrated Initial Position

![Image](b) Demonstrated Apex

![Image](c) Planned Initial Position

![Image](d) Planned Apex

Figure 2: Low-Velocity Pendulum Trajectory

the trajectory pose associated with keyframe $g_{i-1}$. This time
period extends until $t_w$, which is defined as the following:

$$t_w = t_f \cdot \frac{i}{n}$$

(1)

The change in end-effector positions between $d_{j-1}$ and $d_j$
along the $x$, $y$, and $z$ axis are used to determine the veloc-

i
ties at keyframe $g_i$. We assume that the trajectory demon-
stration starts near the first keyframe and passes near the end
keyframe.

Once the geometric and dynamic constraints have been
aligned, the skill is represented by the set $C = \{c_1, ..., c_n\}$,
where $n$ is the number of keyframe poses in $G$, and $c_i$
represents a constraint as a 10-element tuple containing
the geometric constraint (7-DOF arm’s joint state) and dy-
namic constraint (velocities along the $x$, $y$, and $z$ axis). An
acceleration-limited planner, described in (Kunz and Stil-
man 2014b), is then applied to this set of constraints to cre-
a a trajectory that, when executed, satisfies the demonstrated
constraints (Kunz and Stilman 2014a; 2014b).

**Preliminary Results** We applied this approach toward
learning a pendulum skill illustrating the effect of demon-
strated velocity on the robot’s skill execution. Two ver-
sions of the same skill were demonstrated: one in which
the pendulum is tapped at a low speed, and one in which
the pendulum is swung at a higher speed. The velocity at
which the pendulum is tapped is thus reflected in the height
of the pendulum’s swing. The low-velocity demonstration
and resulting trajectory created by the planner are shown
in Figure 2. The higher-velocity demonstration and result-
ing planned trajectory are shown in Figure 3. As expected,
the pendulum swing differed depending on the demonstrated speed, demonstrating one example of within-domain adaptation.

The primary benefit of this method is that rather than encode the full skill, only the constraints necessary to complete the skill are recorded. This provides an abstraction of the skill, as it leaves the specifics of execution up to the planning stage. Thus, we suggest that this approach is also suitable for within-domain transfer as well. For example, closing a laptop and closing a book have similar goals, but may require different methods (trajectories) to achieve that goal. By encoding the demonstrated skill as its goal (reaching the top of the laptop at a particular downward velocity), the skill representation is transferrable to the book-closing target problem as well.

Conclusions and Future Work

We have proposed two classifications of transfer problems: cross-domain transfer and within-domain adaptation. We have described two approaches to representing skills such that adaptation and transfer are enabled. Trajectory-based skill representations may enable within-domain adaptation by mimicking the teacher’s method of achieving the goal, while goal-based skill representations may be applicable to both cross-domain transfer and within-domain adaptation by imitating the intention of the teacher’s actions, rather than the teacher’s method of reaching the goal.

Future work will involve a study of each skill representation’s adaptability and transferability. Examples of potential adaptation problems include a closing skill in which the representation must be transferred from a box-closing problem to a book-closing problem, and a pushing skill in which the representation must be transferred from pushing a cart to pushing a drawer.

Acknowledgements

We thank Tobias Kunz for applying the acceleration-limited planner to skill constraints to generate the trajectories shown in Figures 2 & 3. This work was supported by ONR Grant #N000141410120 and NSF GRF Grant #DGE-1148903.

References


