An Integrated System for Learning Multi-Step Robotic Tasks from Unstructured Demonstrations

Scott Niekum
University of Massachusetts Amherst
Amherst, Massachusetts
sniekum@cs.umass.edu

Abstract

We present an integrated system for segmenting demonstrations, recognizing repeated skills, and generalizing multi-step tasks from unstructured demonstrations. This method combines recent work in Bayesian nonparametric statistics and learning from demonstration with perception using an RGB-D camera to generalize a multi-step task on the PR2 mobile manipulator. We demonstrate the potential of our framework to learn a large library of skills over time and discuss how it might be improved with additional integration of components such as active learning, interactive feedback from humans, and more advanced perception.

Introduction

A simple system that allows end-users to intuitively program robots is a key step in getting robots out of the laboratory and into the real world. Although in many cases it is possible for an expert to successfully program a robot to perform complex tasks, such programming requires a great deal of knowledge, is time-consuming, and is often task-specific. In response to this, much recent work has focused on robot learning from demonstration (LfD) (Argall et al. 2009), where non-expert users can teach a robot how to perform a task by example. Such demonstrations eliminate the need for knowledge of the robotic system, and in many cases require only a fraction of the time that it would take an expert to design a controller by hand.

Ideally, an LfD system can learn to perform and generalize complex tasks given a minimal number of demonstrations without requiring knowledge about the robot. Much LfD research has focused on the case in which the robot learns a monolithic policy from a demonstration of a simple task with a well-defined beginning and end. This approach often fails for complex tasks that are difficult to model with a single policy. Thus, structured demonstrations are often provided for a sequence of subtasks, or *skills*, that are easier to learn and generalize than the task as a whole, and which may be reusable in other tasks.

However, a number of problems are associated with segmenting tasks by hand and providing individual skill demonstrations. Since the most natural way to demonstrate a task is by performing it continuously from start to finish, dividing a task into component skills is not only time-consuming, but often difficult—an effective segmentation can require knowledge of the robot's kinematic properties, internal representations, and existing skill competencies. Since skills may be repeated within and across tasks, defining skills also requires qualitative judgements about when two segments can be considered a single skill, or in deciding the appropriate level of granularity at which to perform segmentation. Users cannot be expected to manually manage this collection of skills as it grows over time. Given these difficulties, it is unlikely that LfD methods will scale in the long-term if users are required to manually segment demonstration data.

For this reason, we aim to automate the segmentation process. We identify four key issues that are critical to any system that aims to learn increasingly complex tasks from unstructured demonstrations. (By *unstructured*, we refer to demonstrations that are unsegmented, possibly incomplete, and may come from multiple tasks or skills.) First, the robot must be able to recognize repeated instances of skills and generalize them to new settings. Second, segmentation should be able to be performed without the need for *a priori* knowledge about the number or structure of skills involved in a task. Third, the robot should be able to identify a broad, general class of skills, including object manipulation skills, gestures, and goal-based actions. Fourth, the representation of skill policies should be such that they can be improved through practice.

Although some of these issues have previously been addressed individually, no system that we are aware of has jointly addressed them all in a principled manner. We argue that learning from unstructured demonstrations (LfUD) not only demands an integrated solution, but that many of the deepest research questions involved are in fact systems integration issues. First, we introduce an integrated LfUD framework that combines recent work in Bayesian nonparametric statistics and LfD with perception using an RGB-D camera to learn a multi-step task on the PR2 mobile manipulator. Segmentation is achieved using a Beta-Process Autoregressive HMM (Fox et al. 2009), while Dynamic Movement Primitives (Ijspeert, Nakanishi, and Schaal 2003), coordinate frame inference, and simple perception are used to address LfD, policy representation, and generalization. We then discuss the integration challenges that arose when designing this system and examine what additional components may be required in the future to advance the state-of-the-art in LfD.

Learning from Unstructured Demonstrations

We now introduce a framework which has four major capabilities critical for the robust learning of complex tasks from unstructured demonstrations. First, the robot must be able to recognize repeated instances of skills and generalize them to new settings. Given a set of demonstrations for a task, we use a Beta-Process Autoregressive Hidden Markov Model (BP-AR-HMM) (Fox et al. 2009) to parse the demonstrations into segments that can be explained by a set of latent skills, represented as vector autoregressive (VAR) processes, a special case of linear dynamical systems. The BP-AR-HMM enables these skills to be shared across demonstrations and tasks by employing a feature-based representation in which each skill corresponds to a feature that may or may not be present in a particular trajectory. Furthermore, this representation allows each trajectory to transition between skills in a unique manner, so that skills can be identified flexibly in a variety of situations, while still retaining globally shared properties.

Segmentation of trajectories into VAR models allows for tractable inference over the time-series dependencies of observations and provides a parameterization of each skill so that repeat instances can be recognized. This representation models how state changes over time, based on previous state values, potentially allowing instances of the same underlying skill to be recognized, even when performed with respect to different coordinate frames. The BP-AR-HMM also models skill-dependent noise characteristics to improve the identification of repeated skills. By recognizing repeated skills, a skill library can be incrementally constructed over time to assist in segmenting new demonstrations. Additionally, skill controllers that have been previously learned and improved through practice can be reused on new tasks. Thus, recognition of repeated skills can reduce the amount of demonstration data required to successfully segment and learn complex tasks. Similarly, if we have multiple examples of a skill, we can discover invariants that allow us to generalize the skill to new situations robustly. In this paper, we use this data to identify the coordinate frames that each skill takes place in, as described in detail in the next section.

Second, segmentation must be able to be performed without the need for *a priori* knowledge about the number or structure of skills involved in a task. The BP-AR-HMM places a beta process prior over the matrix of trajectory-feature assignments, so that a potentially infinite number of skills can be represented; the actual finite number of represented skills is decided upon in a principled, fully Bayesian way. Skill durations are modeled indirectly through a learned self-transition bias, preventing skills from being over-segmented into many small components unnecessarily. The BP-AR-HMM also provides reliable inference, having only a few free parameters that are robust to a wide range of initial settings and hyperparameters that conform to the data as inference progresses. Thus, little tuning should be necessary for varying tasks for a given robotic platform.

Third, our system must be able to identify a broad, general class of skills. Since our segmentation method is based upon state changes, rather than absolute state values, we are able to identify a wide array of movement types ranging from object manipulation skills to gestures and goal-based actions. Furthermore, by identifying the relevant coordinate frame of repeated skills, we can discover specific objects and goals in the world that skills are associated with.

Fourth, the representation of skill policies should be such that they can be easily generalized and improved through practice. To accomplish this, we represent skill controllers in the Dynamic Movement Primitive (DMP) framework (Ijspeert, Nakanishi, and Schaal 2003). The spring-damper mechanics of a DMP allow for easy generalization, since the start and goal set-points can be moved, while still guaranteeing convergence and maintaining the "spirit" of the demonstration through the output of the nonlinear function.

Methodology

Demonstrations

We use a Willow Garage PR2 mobile manipulator and the ROS framework for our experiments. Task demonstrations are provide through kinesthetic demonstrations, whereby the teacher physically manipulates the robot to complete the task. At the beginning of each demonstration, the robot captures a depth image and object positions are determined by a visual fiducial placed on each object of interest. Once the demonstration begins, data are collected by recording the 7 joint angles in the left arm and the gripper state (a scalar indicating its degree of closed-ness). Offline, the joint angles are converted to a series of 3D Cartesian positions and 4D quaternion orientations, which are subsampled down to 10 Hz and smoothed, along with the gripper positions.

Perception

To determine the identities and poses of objects in the environment, we use AR tags, a type of visual fiducial. We place a unique AR tag on (or relative to) each object of interest in the environment and use the Alvar AR tag tracking library[1] to estimate the pose of the tag from camera images. Additionally, we integrate depth data from a Microsoft Kinect RGB-D camera to further refine the tag's pose by fitting a plane to points belonging to the tag.

Segmentation

The BP-AR-HMM addresses some of the shortcomings of a standard Hidden Markov Model. It uses a beta process prior that leverages an infinite feature-based representation, in which each time series can exhibit a subset of the total number of discovered modes and switch between them in a unique manner. Thus, a potentially infinite library of modes can be constructed in a fully Bayesian way, in which modes are flexibly shared between time series, and an appropriate number of modes is inferred directly from the data, without the need for model selection. Second, the BP-AR-HMM is

[1] `http://virtual.vtt.fi/virtual/proj2/multimedia/index.html`

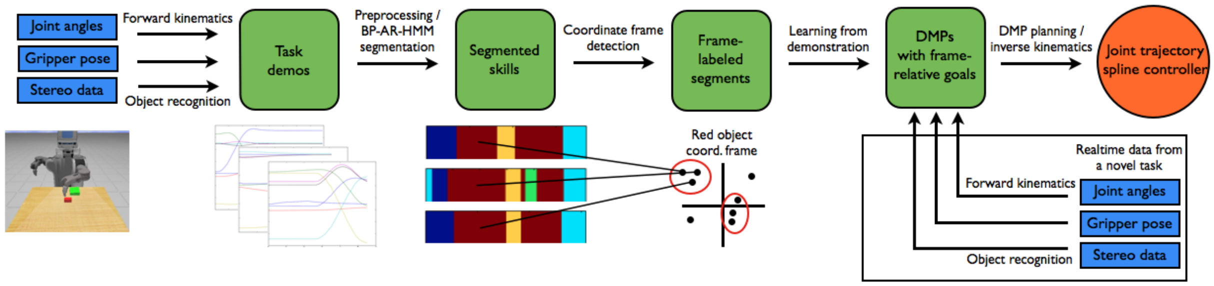


Figure 1: Overview of the framework used in the experiments, as described in section

autoregressive and can describe temporal dependencies between continuous observations as a VAR process, a special case of a linear dynamical system.

We build on a BP-AR-HMM implementation made available by Emily Fox[2] to segment sets of demonstration trajectories. Using similar parameters as in Fox et al. (Fox et al. 2011), we run the sampler 10 times for 1000 iterations each, producing 10 segmentations. Qualitatively, the segmentations across runs were very consistent, but to ensure good results, the segmentation from the 10 runs with the highest log likelihood of the feature settings is selected. These segmentations provide a discretization of complex continuous demonstration data into statistical motion categories that we can use as skills for building complex task behaviors. We can then look for invariants amongst the examples of each motion category to give each skill more semantic meaning in the context of the task; specifically we look for relevant coordinate frames that help describe the behavior of each skill across multiple examples.

Coordinate Frame Detection

After the demonstrations are segmented, each segment is examined to infer the coordinate frame that it is occurring in. Even though segments assigned to the same skill correspond to similar movements, they may be happening in different frames of reference. For example, a repeated reaching motion may be classified as being generated by the same skill, but be reaching toward several different objects. In order to robustly replay tasks in novel configurations, it is desirable to determine which coordinate frame each segment is associated with so that DMP goals can be generalized correctly.

We define a coordinate frame centered on each known object, along with one centered at the torso of the robot. Other frames could be used as well if desired, such as a frame relative to the gripper, or a world frame. Then, the final point of each segment is plotted separately in each of the coordinate frames, and clusters are found in each frame by identifying points within a Euclidean distance threshold of each other. The reasoning is that clusters of points indicate that multiple segments have similar endpoints in a particular coordinate frame, suggesting that the skill often occurs in that frame of reference.

[2] `http://stat.wharton.upenn.edu/~ebfox/software`

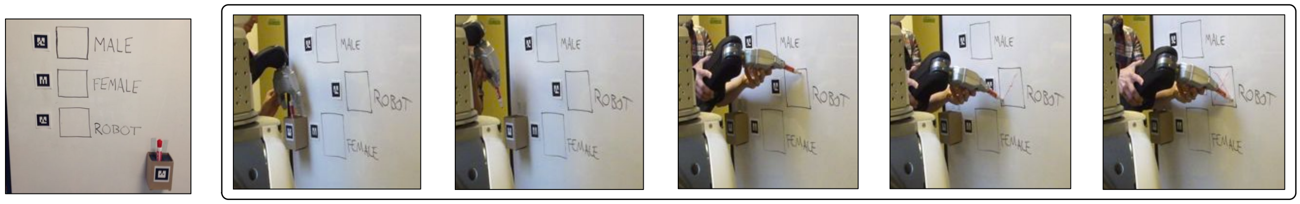
After the points are clustered in each frame, all the singleton clusters are discarded. If any remaining segment endpoint belongs only to a cluster in a single coordinate frame, then the evidence is unambiguous, and that segment is assigned to that coordinate frame. Otherwise, if a segment endpoint belongs to clusters in multiple frames, it is simply assigned to the frame corresponding to the largest cluster. It should be emphasized that the any coordinate frame inference method could be used in place of ours, and that there are many other skill invariants that could be exploited. The purpose of this method is primarily to demonstrate the utility of being able to segment and recognize repeated skills.

Task Replay

To perform a task in a novel configuration, we first determine the poses and identities of objects in the scene, using the visual fiducials. The position of each object is then examined to find the demonstration that begins with the objects in a configuration that is closest to the current one in a Euclidean sense. We only consider demonstrations that have an identified coordinate frame for every segment, so that the task will generalize properly. A DMP is then created and trained using a simple regression method (Ijspeert, Nakanishi, and Schaal 2003) for each segment in the demonstration. However, rather than using the final point of a segment as the goal of a DMP, each goal is adjusted based on the coordinate frame that the segment takes place in. If the segment is associated with the torso frame, it requires no adjustment. Otherwise, if it is associated with an object frame, the goal is adjusted by the difference between the object's current position and its position in the demonstration. Finally, the DMPs are executed in the sequence specified by the demonstration. A plan is generated by each of the DMPs until the predicted state is within a small threshold of the goal. Each plan is a Cartesian trajectory (plus a synchronized gripper state) that is converted into smooth joint commands using inverse kinematics and spline interpolation. A graphical overview of our method is shown in Figure 1. For more technical details on how our system works, refer to earlier work by Niekum et al.(Niekum et al. 2012)

Experiments

We now demonstrate the effectiveness of our method on a task using the PR2 mobile manipulator. Figure 2(a) shows



(a) Task example (b) A kinesthetic demonstration

Figure 2: The whiteboard survey task.

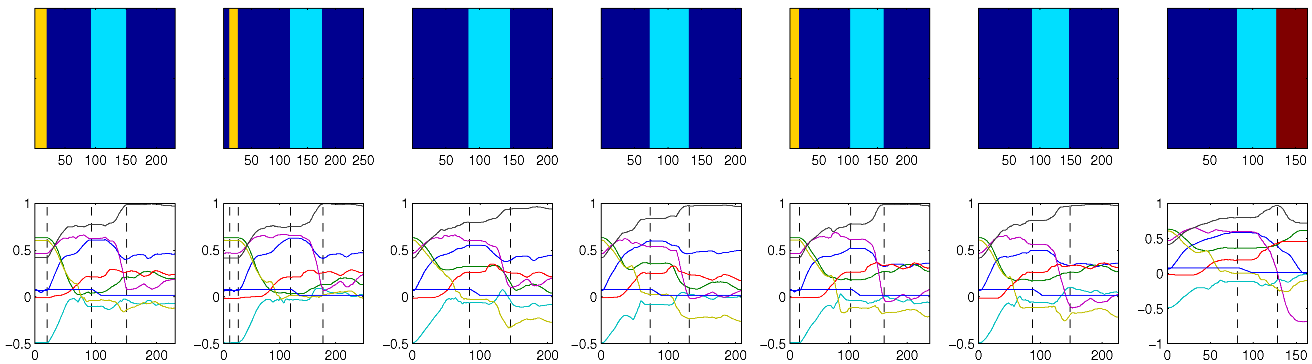


Figure 3: Top: BP-AR-HMM segmentations of the 7 demonstration trajectories for the whiteboard survey task. Time (in tenths of a second) is shown on the x-axis. Skill labels at each time step are indicated by unique colors. Bottom: Segmentation points overlaid on the demonstrated 8D movement data.

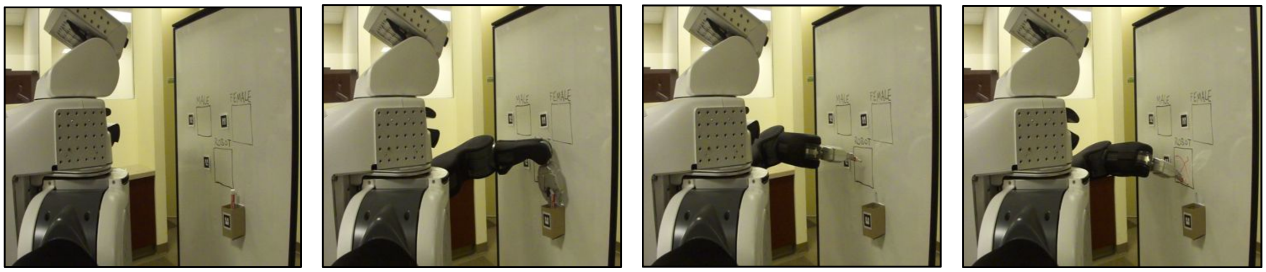
one configuration of the task in which the PR2 must fill out a survey on a whiteboard by picking up a red marker and drawing an ‘X’ in the checkbox corresponding to “robot” while ignoring the checkboxes for “male” and “female”. Each checkbox has its own unique fiducial placed one inch to the left of it, while the container that holds the marker has a fiducial directly on its front. The positions of the checkboxes and the marker container on the whiteboard, as well as the position of the whiteboard itself, change between task configurations. Two kinesthetic demonstrations in each of three task configurations were provided, along with one additional demonstration in which the marker is picked up and then lifted above the robot’s head. An example demonstration is shown in Figure 2(b).

Figure 3 shows that the BP-AR-HMM generally parses the demonstrations into three main segments, corresponding to reaching for the marker, grasping and lifting the marker, and drawing an ‘X’ in the checkbox. However, the reaching and drawing segments are considered to be the same skill. This appears to happen because both motions are statistically similar, not in terms of absolute position, but in the way that the positions evolve over time as a VAR system. Our coordinate frame detection successfully disambiguates these skills and splits them into two separate skill/coordinate frame combinations. Demonstrations 1, 2, and 5 contain a small additional skill near the beginning that corresponds to a significant twitch in the shoulder joint before any other movement starts, which appears to correspond to the teacher’s first contact with the arm, prior to the demonstration. Finally, although the last demonstration is of a different task, the reaching and grasping/lifting skills are still successfully recognized, while the final motion of lifting the marker over the robot’s head is given a unique skill of its own. Despite having only a single example of the overhead skill, the BP-AR-HMM robustly identified it as being unique in 50 out of 50 trial segmentations, while also recognizing other skills from the main task. After the learning phase, the robot was able to successfully replay the task in three novel configurations, an example of which is shown in Figure 4.

Integration Issues

We now highlight several of the integration issues we faced while developing this system that may be general to many integrated robot learning systems. It is worth noting that some of the issues (particularly the first two) turned out to be deep research questions that we had not originally planned to address; the process of trying to build an integrated system revealed and motivated these new, important questions that generally do not come up in stand-alone systems.

1. **Finding common representations:** When building integrated systems, designers must find common representations that allow pieces of system to fit together without the use of ad-hoc heuristics. By reasoning at the level of *movements* in our system, we gain a common representation to reason about data that can originate from diverse sources such as planners, human demonstrations,



(a) Starting pose (b) Reaches for marker (marker frame) (c) Grasps marker and lifts toward checkbox (robot checkbox frame) (d) Draws 'X' (robot checkbox frame)

Figure 4: Successful task replay on a novel test configuration for the whiteboard survey task, demonstrating generalization. From left to right: the starting pose and the final point of each executed DMP. Automatically detected coordinate frames used for each segment are listed in parentheses.

and hand-coded controllers. Trying to integrate each of these types of data (plans, motion capture data, and code, respectively) in their native format would be much more difficult. Furthermore, by learning at the movement level, we can leverage DMPs, which provide a natural mechanism for generalization and improvement via reinforcement learning, without many of the trappings of traditional state-space representations of more general policies.

2. **Robustness:** Integrated systems must find ways to cope with the brittleness of any particular piece much more so than isolated systems, as the the failure of any given piece can cause the entire system to fail. Thus, as systems grow in size, failure rates will quickly increase unless component failures can be handled gracefully. In our work, we had to be robust to poor segmentations, perceptual errors, and planned trajectories that are physically infeasible. Through filtering, multiple segmentations, and relevant segment selection, we were able to handle these failure modes and produce repeatable results.

3. **Stand-in components:** Another challenge is identifying which components of a system can be replaced with "stand-ins" and which ones must be implemented in a principled manner. In our case, we used visual fiducials as a stand-in for a complete perception system that can recognize objects in a natural scene and calculate their pose. Conversely, in previous research, the use of manual segmentations or segmentation heuristics has prevented LfD from generalizing in many important ways; therefore, we chose a principled (albeit complex) solution for segmentation—the BP-AR-HMM.

4. **Software design:** Due to the tightly coupled nature of integrated systems, it can often be difficult to design independent software modules for each system component. However, as the size of these systems grow, it becomes increasingly important to do so for code maintainability and reusability. This is especially important for complex robotic systems. given that each component may encapsulate years or research effort and code refinement. The use of ROS partially addresses these issues by providing a communications layer between components, consistent organizational principles, and a standard method for sharing code. We have released our DMP implementation and perception code as ROS stacks, both of which are robot-agnostic.

Discussion

We presented a method for segmenting demonstrations, recognizing repeated skills, and generalizing complex tasks from unstructured demonstration. This system was largely an integration effort, bringing together many existing techniques into a coherent system that has state-of-the-art learning capabilities. This leads us to ask: how far might integration efforts alone take robot learning? The experimental results presented indicate that we may already have the necessary tools to take robot learning much further that ever before, and that many of the primary remaining challenges are integrative in nature. With that in mind, we now discuss some possible directions for future integration efforts to expand the capabilities of our system.

Rather than replaying a skill sequence from a single demonstration that is deemed most relevant, it would be better to intelligently sequence skills on-the-fly. Segmented demonstrations could be used to build a high-level representation of the task such as a finite state machine of skills, while any number of off-the-shelf classification methods might be used to model skill initiation and termination conditions that guide the skill transitions. This would allow the robot to better cope with novel situations, be more reactive to the environment, incrementally learn a more complete description of the task, and deal with possible contingencies that can arise during execution. Additionally, users could be allowed to provide interactive corrections to the robot during task replay as an additional way to incrementally update the task structure and refine skills.

It is also desirable to have the robot be able to learn in the absence of new demonstration data, either through sparse human feedback, or entirely autonomously. Simple natural language feedback and reinforcement learning could be used to improve skills in this manner. Active learning could also be employed to allow the robot to learn more about the world autonomously. For example, to learn more about an object's affordances, the robot could choose a skill to execute on (or with) the object based on which skills have been used on other similar objects. This similarity could be based on affordances shared between objects, or could even integrate perception as well, through a visual comparison.

These are just a handful of the ways in which our system could be augmented to make it more powerful. In general, these ideas do not require new theory or machine learning techniques, but rather solid application of existing methods and integration with the rest of the system. We will have to solve integration problems such as:

- How can we continue to leverage skill learning progress if skill segmentations shift when more data becomes available?
- How will poor and changing segmentations impact building a high-level task representation?
- How can reinforcement learning make progress if skill initiation and termination conditions are imperfectly modeled?
- How can demonstration data and interactive human feedback be integrated?
- What are the connections between perceptual data, world models, and intelligent skill sequencing?

If integration questions like these can be answered, then we claim that by using current control, LfD, and perception algorithms, it is possible to build a robotic learning system that is far more capable than the current state-of-the-art. In the future, we hope that this work can be a building block of a end-to-end LfD system that allows end-users to naturally program robots to perform useful, complex tasks in home and workplace.

References

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

Fox, E.; Sudderth, E.; Jordan, M.; and Willsky, A. 2009. Sharing features among dynamical systems with beta processes. *Advances in Neural Information Processing Systems 22* 549–557.

Fox, E.; Sudderth, E.; Jordan, M.; and Willsky, A. 2011. Joint modeling of multiple related time series via the beta process. *arXiv:1111.4226*.

Ijspeert, A.; Nakanishi, J.; and Schaal, S. 2003. Learning attractor landscapes for learning motor primitives. *Advances in Neural Information Processing Systems 16* 1547–1554.

Niekum, S.; Osentoski, S.; Konidaris, G.; and Barto, A. 2012. Learning and generalization of complex tasks from unstructured demonstrations. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*.