Mining Affordances for Grasping and Manipulation

Stephen Hart, Roderic Grupen, David Jensen
Computer Science Department
University of Massachusetts Amherst
{shart, grupen, jensen}@cs.umass.edu

1 Introduction

In order to learn generalized knowledge in grasping and manipulation tasks, a robot must autonomously discover the affordances [1] of manipulable objects in the world. This poster presentation proposes a methodology in which salient sensorimotor features are *mined* from training experience. As a result, actions and affordances can be conditioned on task success. This technique allows the robot to learn generalizable knowledge about how to perform manipulation tasks in real-world environments in a robust manner, and does not rely on prior knowledge geometric planning methods.

Task knowledge can be decomposed into *declarative* and *procedural* components [3]. The declarative structure captures abstract knowledge about the task; e.g. to pick up an object, we must first find the object, reach to it, and then grasp it. The procedural structure captures knowledge about how to instantiate the abstract policy in a particular setting; e.g. in this case, we must use our left hand to pick up the object and use an enveloping grasp. With such a decomposition, it is possible to represent task knowledge that is transferable across various environmental contexts. The declarative structure of a task defines an abstract schema that can guide an agent's behavior in the world, while the procedural structure decorates this schema with resources appropriate for the given context.

2 Learning Task Expertise Developmentally

It is possible to learn both declarative and procedural knowledge for manipulation tasks in a developmental way. By constraining the environmental context and the choice of resources, the robot can learn through trial-and-error how to sequence actions to accomplish the task. As a result, an initial estimate of the declarative structure can be found. For example, consider a robot learning how to pick up objects. If a small cube is placed close to its right arm, it can learn very quickly to Localize, then Reach, and then Grasp, if its only choice of resources are a stereo camera pair, its right arm, and 2 fingers, respectively. This sequence represents declarative knowledge about how to pick up objects, regardless of what resources were used in training.

This abstract policy can the be used as a starting point when the robot tries to accomplish the same task in a different setting and with more resources available to it. If the robot encounters new objects that are placed in varying locations in front of it, the robot can explore other resource bindings (both left and right arms, 3 finger grasps, etc.) that might work well in the new context, while following the abstract policy of Localize-Reach-Grasp. This exploration phase allows the robot to learn more complete procedural knowledge about a task after a bootstrapping phase in which it learns the abstract policy.

2.1 Learning the Declarative Structure

Previous work using the *Control Basis* approach [2] has framed the problem of learning task knowledge as a Markov Decision Process which can be solved using reinforcement learning techniques such as Q-Learning. The experiments developed in this work allowed a robot to learn the declarative structure of a task in a constrained setting through trial-and-error experimentation.

2.2 Learning the Procedural Structure

We propose using probabilistic models of relational data to learn the procedural structure of a task. In particular, we will use *Relational Dependency Networks* (or RDNs) [4] to find the statistical dependencies between observable sensorimotor variables and task success. Relational models are useful because they provide a framework for learning from experience in a variety of settings. In this type of model, random variables are designated as attributes. Sets of attributes are gathered into data objects that are related through the structure of the data. It is the statistical correlation between the attributes that is learned using the RDN algorithm.

We can use RDNs to capture task knowledge by defining each data object to correspond to an abstract control action (e.g. Localize, Reach, Grasp), each of which is designed to be asymptotically stable in the sense of Lyapunov [2]. These objects have attributes pertaining to "convergence state," "resource binding," as well as any features that can be observed through the execution of the corresponding controller. For example, the Localize controller might have attributes pertaining to the Cartesian positions of all objects in the camera's field of view. This methodology has been used to learn how to successfully pick up a class of objects with a humanoid robot, based on attributes such as locale and retinal scale (Figure 1).

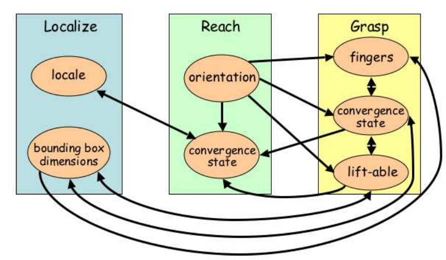

Figure 1: PickUp RDN built from experimental data

In general, the number of attributes available for a controller may be infinite. A Localize controller, for example, may gather information from a pair of stereo images. Because there are an infinite number of possible features available from an image (e.g. all Gaussian filters at all scales and orientations), it may be necessary to sample from the full set, resampling if poor policies are learned.

This poster presents a developmental way of learning procedural and declarative knowledge about picking up objects along with experiments on a humanoid robot following this methodology.

References

[1] J. Gibson, "The theory of affordances," in *Perceiving, acting and knowing: toward an ecological psychology*. Hillsdale, NJ: Lawrence Erlbaum Associates Publishers, 1977, pp. 67–82.

[2] M. Huber, "A hybrid architecture for adaptive robot control," Ph.D. dissertation, Department of Computer Science, University of Massachusetts Amherst, 2000.

[3] J. M. Mandler, *The Foundations of Mind: Origins of Conceptual Thought*. New York, NY: Oxford University Press, 2004.

[4] J. Neville and D. D. Jensen, "Collective classication with relational dependency networks," in *2nd Workshop on Multi-Relational Data Mining, KDD*, 2003.