Natural Methods for Learning and Generalization in Human-Robot Domains

Monica N. Nicolescu and **Maja J. Matarić**
Computer Science Department
University of Southern California
941 West 37th Place, Mailcode 0781
Los Angeles, CA 90089-0781
monica|mataric@cs.usc.edu

Abstract

Human-robot interaction is a growing research domain; there are many approaches to robot design, depending on the particular aspects of interaction being focused on. The challenge we address in this paper is to build robots that have the ability to learn to perform complex tasks and refine their acquired capabilities through interaction with humans in the environment. We propose an approach for teaching robots by demonstration similar to the one people use among themselves: demonstrate a task, then allow the learner to perform it under the teacher's supervision. Depending on the quality of the learned task, the teacher may either demonstrate the task again or provide specific feedback during the learner's practice trial. Thus, *generalization* over several demonstrations and using *feedback* during practice execution trials are key capabilities for refining previously learned task representations. We validated these concepts with a Pioneer 2DX mobile robot learning various tasks from demonstration.

Introduction

Robots that can successfully and efficiently interact with humans require adaptation and learning capabilities for most non-trivial interactions. This enables the robots to not only adapt and improve their performance, but also to be more accessible to a larger range of users, from the lay to the skilled.

Designing controllers for robotic tasks is usually done by people specialized in programming robots. Even for them, most often, this is a complicated process, and it essentially requires creating by hand a new and different controller for each particular task. Although certain parts of controllers, once refined, can be reused, it is still necessary to, at least partially, redesign and customize the existing code for each new task. If robots are to be effective in human-robot domains, even users without programming skills should be able to interact with them and "re-program" them.

Thus, *automating the robot controller design process* becomes of particular interest. A natural approach to this problem is the use of *teaching by demonstration*. However, the majority of robot teaching approaches to date has been focused on learning policies (Hayes & Demiris 1994; Schaal 1997), or temporally extended sensory-motor skills (Demiris & Hayes 2002). Techniques for learning complex task structures have also been presented (Kuniyoshi, Inaba, & Inoue 1994), but they are highly sensitive to the structure of the environment and of the demonstration, and do not allow for further improvement or adaptation if the task is not learned correctly in the first trial.

Our goal is to develop a flexible mechanism that allows a robot to learn and refine representations of *high level tasks, from interaction with a human teacher, based on a set of underlying capabilities (behaviors) already available to the robot*.

An important challenge for a learning method that is based on robot's observations is to distinguish between the relevant and irrelevant information being perceived. Also, the limited sensing capabilities of the robot could prevent it from observing actions relevant to the task. Thus, in order to learn a task correctly in such conditions, the teacher needs to provide the robot additional information beyond the demonstration experience. Therefore, it is essential that the robot have the ability to refine previously learned task representations, and to incorporate the additional information provided by the teacher.

To address these challenges we propose an approach similar to what people use when teaching each other through showing. Humans typically demonstrate a task once and then supervise a set of practice trials in which the learner performs what was learned. If needed, during these runs the teacher provides *feedback cues* to indicate corrections (irrelevant actions or missing parts of the task). Alternatively, the teacher may also provide several new demonstrations that the learner could use for *generalization*. Our overall strategy for learning and refining task representations is presented in Figure 1. In the following sections we describe our method for learning and refining task representations through generalization and feedback.

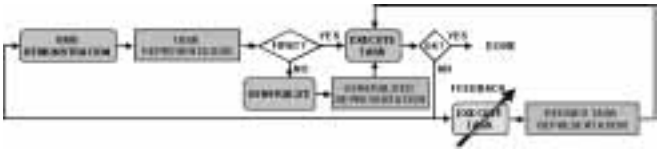


Figure 1: Learning and refining tasks through demonstrations, generalization and teacher feedback

Learning through demonstration

Control Architecture - Behavior Networks

As underlying control architecture we use an extension of the standard Behavior-Based System we developed, in which tasks are represented as networks of *abstract behaviors* (Figure 2) (Nicolescu & Matarić 2002).

In a behavior network, the links between behaviors represent precondition-postcondition dependencies, which can have three different types: *permanent*, *enabling* and *ordering*. Thus the activation of a behavior is dependent not only on its own preconditions (particular environmental states) but also on the postconditions of its relevant predecessors (*sequential preconditions*). More details on this architecture can be found in (Nicolescu & Matarić 2002).

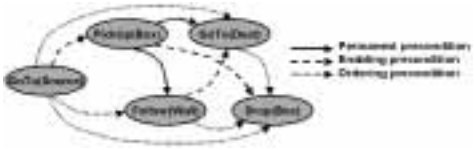


Figure 2: Example of a behavior network for an "Object transport" task

Within this architecture, behaviors are build from two components: one related to perception (*Abstract behavior*), the other to action (*Primitive behavior*). The *abstract behavior* is an explicit specification of the behavior's activation conditions (i.e., preconditions), and its effects (i.e., postconditions). The behaviors that do the work that achieves the specified effects under the given conditions are called *primitive behaviors*. An *abstract behavior* takes sensory information from the environment and, when its preconditions are met, activates the corresponding *primitive behavior(s)*, which achieve the effects specified in its postconditions.

Learning task representations

The most common approach for robot teaching is through the use of demonstrations, the same strategy we are also going to use as the main modality for instruction. Two different methods for learning from demonstration exist: *learning by observation*, in which the learner passively observes the teacher performing a task, and *learning by experience*, in which the robot performs the task along with the teacher during the demonstration.

In our particular approach to learning, we use *learning by experienced demonstrations*. This implies that the robot actively participates in the demonstration provided by the teacher, by following the human and experiencing the task through its own sensors. This is an essential characteristic of our approach, and is what is providing the robot the data necessary for learning. In the mobile robot domain the experienced demonstrations are achieved by following and interacting with the teacher. The advantage of *putting the robot through* the task during the demonstration is that the robot is able to adjust its behaviors (through their parameters) using the information gathered through its own sensors. In addition to experiencing parameter values directly, the execution of the behaviors provides observations that contain temporal information for proper behavior sequencing, which would be tedious to design by hand for tasks with long temporal sequences.

Irrespective of the demonstration strategy being used, an important challenge for these learning methods is to distinguish between the relevant and irrelevant information being perceived. Putting the entire responsibility on the learner to decide between relevant and irrelevant observations, such as when learning solely by observation, increases the complexity of the problem and leads to more complicated, sometimes ineffective solutions. During demonstrations humans almost always make use of additional simple cues and instructions that facilitate the learning process and bias the learner's attention to the important aspects of the demonstration (e.g. *"watch this"*, *"lift that"*, etc.). Although simple, these cues have a large impact on the robot's learning performance: by relating them with the state of the environment at the moment when they are received, the learner is provided with information that may otherwise be impossible or extremely hard to obtain only from the observed data.

Therefore, in order for a robot to learn a task effectively, the teacher also needs to provide it with additional information beyond the perceived demonstration experience. To achieve this, we add *verbal instruction* to the existing demonstration capabilities of our system. With this, the teacher can provide the following types of information:

- **"HERE"** - indicates moments in time during the demonstration when the environment presents aspects that are relevant for the task. These indications are general (simple hints meaning *"pay attention now"*) and by no means spell out for the robot the representation of the presented task. While such indications allow the robot to distinguish some of the irrelevant observations, they may still not help it to perfectly learn the task. For this, generalization techniques and feedback-practice runs will be applied.
- **"TAKE", "DROP"** - instructions that induce the robot into performing certain actions during the demonstration (in this case **Pick Up** and **Drop** small objects), actions that would be otherwise impossible to trigger in a teacher-following-only learning approach. In our case, we instruct the robot to open and close its gripper, when the task to be learned involves moving certain objects around.
- **"START", "DONE"** - instructions that signal the beginning and the end of a demonstration, respectively.

For our work, we focus on a strategy that would help a robot build a high-level task representation of a more complex, sequentially structured task, built from the existing behavior set. We do not attempt to reproduce exact trajectories or actions of the teacher, but rather learn the task in terms of its high-level goals. The general idea of our learning mechanism is to create a mapping between the observations gathered during the demonstration and the robot's existing skills that achieve the same observed effects. During a demonstration the robot adds to the network task representation an instance of all behaviors whose postconditions have been detected as true, and during which there have been relevance signals from the teacher, in the order of their occurrence (on-line stage). At the end of the teaching experience, the intervals of time when the effects of each of the behaviors have been true are known, and are used to determine if these

effects have been active in overlapping intervals or in sequence. Based on the above information, the algorithm generates proper dependency links between behaviors (i.e., *permanent*, *enabling* or *ordering*) (off-line stage). This one-shot learning process and a large set of learning experiments are described in more detailed in (Nicolescu & Matarić 2001).

Generalization from several demonstrations

For a teaching by demonstration approach to be effective, it is essential that the robot learn from as few demonstrations as possible. A robot house keeper is of little use if the owner must show it hundreds of times how to bring in the mail. Therefore, statistical learning techniques, which rely on a large number of training examples, are not appropriate for our desired approach.

Given the directed acyclic graph (DAG)-like structure of the behavior network representation of the robot tasks, we consider the *topological* representation of such a network to be a linked list of behaviors, obtained by applying a *topological sort* on the behavior network graph. By using the topological form of the networks as training examples for our domain, the problem of generalization from multiple demonstrations is equivalent to inferring a regular expression (Finite State Automaton (FSA) representation) from a set of given sample *words* (Figure 3(a)). In this analogy, each symbol in a given *word* corresponds to a behavior in a topological representation.

Unfortunately, applying standard methods for regular expression inference, such as the K-TSI Inference Algorithm (Garcia & Vidal 1990), or Morphic Generator Grammatical Inference (MGGI) (P. Garcia & Casacuberta 1987), to this generalization problem yields results that are too complex (in terms of the obtained FSA representations) even for very simple examples. This is due to the fact that these methods assume that all the training examples are correct and they try to fit them as well as possible. For our robot domain, in which the inaccuracies in the training examples are exactly the problem we need to solve, these methods are therefore not well suited.

In robotics, existing methods for generalization from demonstrated examples are largely based on function approximation (Kaiser 1997). Since our tasks are encoded in graph-like representations, we need a different method for generalizing across them.

As discussed above, the two main inaccuracies that can occur in the learning process are *learning irrelevant steps* (false positives) and *omission of steps that are relevant* (false negatives).

Our approach for generalization is to build a task representation that encodes the specifics of each input example, but most importantly that points out the parts that are common to each of them. As a measure of similarity we consider the longest list of common nodes between the *topological* forms of the sample tasks. Based on this information we further construct a *generalized topology* in which nodes that are common to both tasks will be merged, while the others will appear as alternate paths in the graph. For example, for the examples presented in Figure 3(a), behaviors A, B and F constitute the longest subsequence of common nodes

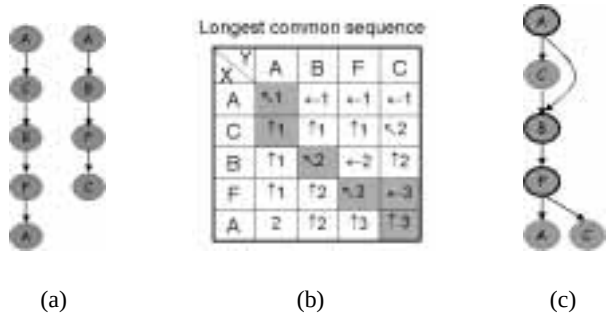


Figure 3: Generalization across multiple demonstrations. (a) Training examples; (b) Longest common sequence table; (c) Generalized topology

(Figure 3(b)). The representation resulted after "merging" the initial graphs at their common nodes is shown in Figure 3(c).

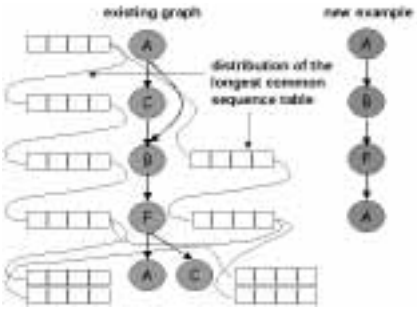


Figure 4: Incorporating new demonstrations: efficient computation of longest common sequences between two tasks.

The generalization process is incremental, meaning that each newly acquired experience is *incorporated* into the existing task representation. This task is represented as a *directed graph*, with alternative paths resulting from previously included examples. Thus, incorporating a new demonstration into the existing structure results in applying the previous algorithm between the new example and all the possible paths of the existing graph. However, since the LCS at a given depth in the graph is given only by paths from the nodes at the higher levels, we can efficiently compute the LCS by distributing the LCS table in the graph, thus avoiding a LCS table computation for each existing path (Figure 4).

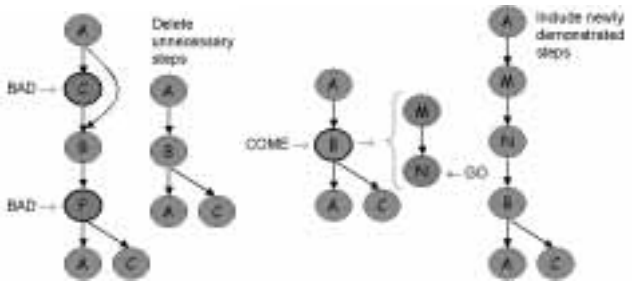
Practice and teacher feedback

Generalization over several training examples helps in identifying the steps that occur most often and that most probably are a part of the task. However, repeated observations of irrelevant steps may inadvertently bias the learner toward including them in the representation. Also, limitations in the sensing capabilities of the robots and particular structures in the environment may prevent the robot from observing steps that are relevant.

The *practice* trials allow the teacher to observe the execution of the robot and to point more accurately to where these

problems occurred. Simple spoken feedback cues allow the teacher to indicate that:

- **"BAD"** - the step that the robot is currently executing is irrelevant for the task. This step is then labeled as irrelevant and is removed from the task representation (Figure 5(a)).
- **"COME"** → **"GO"** the robot has missed relevant steps of the task. In this case, the teacher demonstrates again the missing part of the task, and these steps are incorporated into the task representation (Figure 5(b)).



(a) "BAD" feedback (b) "COME", "GO" feedback

Figure 5: Using feedback for task refinement

Experimental results

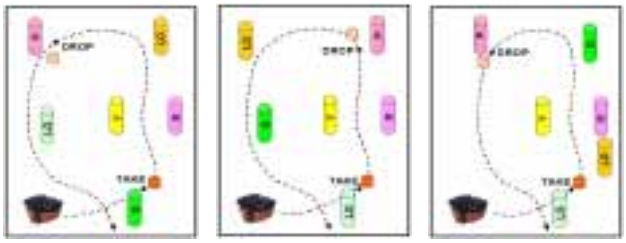
We implemented and tested our concepts on a Pioneer 2-DX mobile robot, equipped with two rings of sonars (8 front and 8 rear), a SICK laser range-finder, a pan-tilt-zoom color camera, a gripper, and on-board computation on a PC104 stack. We performed the experiments in a 5.4m x 6.6m arena. The robot was programmed using AYLLU (Werger 2000), an extension of C for development of distributed control systems for mobile robots. For the voice commands and feedback we used an off-the-shelf Logitech cordless headset, and the IBM ViaVoice software recognition engine.

The robot has a behavior set that allows it to track cylindrical colored targets (**Track(ColorOfTarget, GoalAngle, GoalDistance)**), to pick up (**PickUp(ColorOfObject)**) and drop small colored objects (**Drop**).

Learning by generalization from several examples

We demonstrate the *generalization* abilities of the robot by teaching it an object transport task in three consecutive demonstrations, performed in different environmental setups (Figure 6), and purposely designed to contain incorrect steps and inconsistencies. The next section shows how already learned/generalized tasks can be further refined through *practice* and *feedback*.

The environment consists of a set of cylindrical targets, in colors that the robot is able to perceive. The teacher leads the robot around these targets, while also instructing it when it has to pick up or drop a small orange box. The task to be learned is as follows: go to either the **Green (G)** or the **Light Green (LG)** targets, then pick up an **Orange (O)** box, go between the **Yellow (Y)** and **Red (R)** targets, go to the **Pink (P)** target, drop the box there, then go to the **Light Orange (LO)** target and come back to the target **Light Green**.



(a) First demonstration (b) Second demonstration (c) Third demonstration

Figure 6: Structure of the environment and course of demonstration

The sketched courses of the three demonstrations show that none of them corresponds exactly to the target task. Besides containing unnecessary steps (such as a final visit to a **Green** target in the first trial), these training runs also contain inconsistencies, such as the visits to the **Light Orange** target which happened at various stages during the demonstrations. Figure 7 presents the task representations obtained after each "learning → generalization" process.

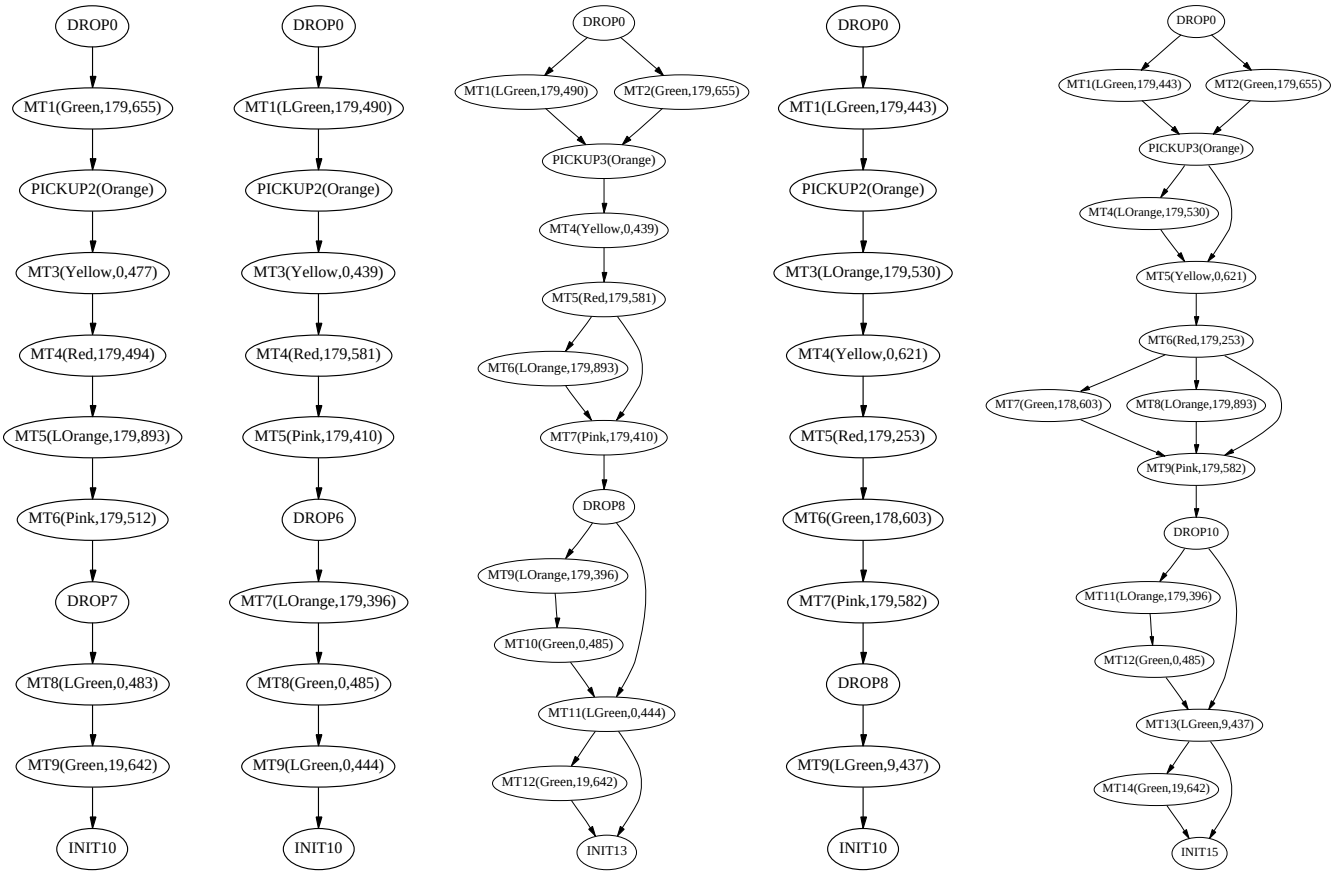
The generalized task representation captures the main structure of the task while at the same time dealing with the irrelevant and inconsistent parts of the demonstrations: both of these situations are captured as becoming a part of bypassed alternate paths which will never be executed. While it is good that the irrelevant actions are thus pruned, the steps demonstrated inconsistently but which are still necessary will have to be included by different means. These results are to be expected: *generalization* alone, when provided with inconsistent examples, is not enough for learning a correct representation.

Learning from practice and teacher feedback

As we saw from the previous section, the generalized network does not yet represent the target task desired by the user. The missing part is a visit to the **Light Orange** target, which should happen right after dropping the box and before going to the **Light Green** target. Since the generalization process already built the remaining of the task structure, simple feedback during a robot practice run is used for refining it to the desired structure.

We performed the practice run in a changed environment, to demonstrate the robustness of our approach. Figure 8(a) shows the robot's sketched trajectory and (dotted) the teacher's intervention, and the type of feedback given to the robot at those times.

An important feature of the practice-feedback approach is the natural characteristic of this process. In order to give the robot appropriate feedback, the teacher doesn't need to know the structure of the task being learned, and thus is shielded from having to know any details about the robot's control architecture. Instead, he simply relies on observing the actions performed by the robot: if they comply with the desired representation, no feedback is given, and if they do not, the corresponding situations are treated with appropriate feedback.



(a) First demonstration (b) Second demonstration (c) Generalized task (d) Third demonstration (e) Generalized task

Figure 7: Evolution of the task representation over three successive demonstrations

(a) Feedback after third demonstration

(b) Feedback after first demonstration

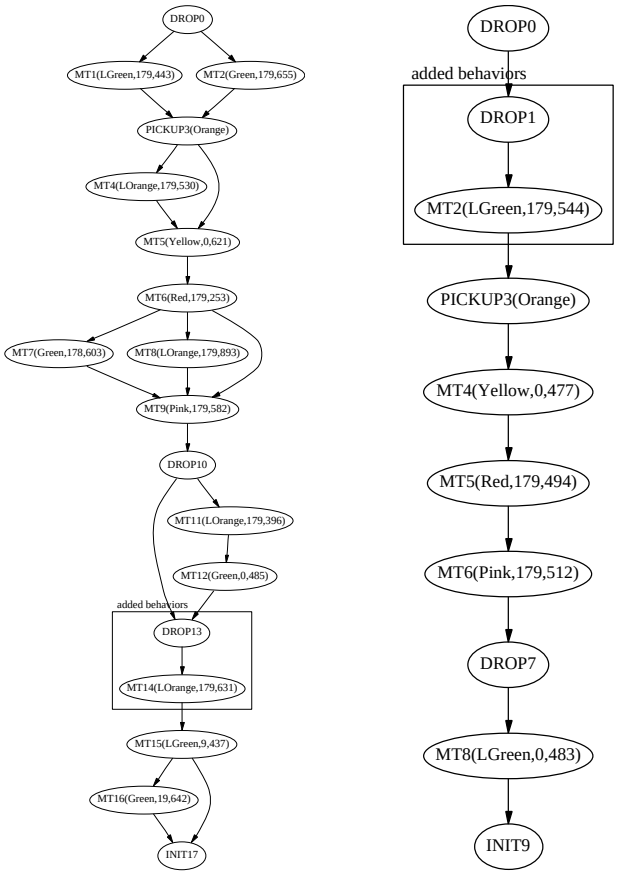
Figure 8: Structure of the environment and course of task execution during practice and feedback

For a different example, we now consider starting from the first demonstration in the previous section. Let us assume that for a second object transport task the teacher considers as wrong the initial visit to a **Green** target, when a **Light Green** target should be visited instead. Also, the visit to the **Light Orange** target is wrong, and not a part of the task as well. Figure 8(b) shows the trajectory of the robot and (dotted) the intervention and messages of the teacher during the robot's practice run. The effects of this feedback are that: the visit to the **Green** target was replaced by a visit to the **Light Green** target, and the visits to the **Light Orange** and **Green** have been removed. Figure 9(b) presents the structure of the task after this practice run.

The *practice-feedback* runs are a much faster and precise method for refining previously learned tasks. Since feedback can be given at any step during the robot's practice, wrong steps can be marked immediately upon observing them being executed, and missing steps can be added as soon as the teacher detects that the robot had skipped to a future step in the task.

Conclusions

Learning capabilities are essential for successful integration of robots in human-robot domains, so robots can learn from human demonstrations and allow for natural interaction with people. Due to inherent challenges of the learning process, it is also important that robots be able to improve their capabilities by receiving additional training and feedback. Toward this end, we presented an approach for teaching by demonstration that enables a robot to learn and refine representa-



(a) After the 3rd demonstration

(b) After the first demonstration

Figure 9: Topologies obtained after practice and feedback

tions of complex tasks. Through generalization, the robot can incorporate several demonstrations of the same task into a unique graph-like representation. Natural cues provided by the teacher through speech allow the robot to further refine this representation.

References

Demiris, Y., and Hayes, G. 2002. Imitation as a dual-route process featuring predictive and learning components: a biologically-plausible computational model. In Dautenhahn, K., and Nehaniv, C., eds., *Imitation in Animals and Artifacts*. MIT Press.

Garcia, P., and Vidal, E. 1990. Inference of k-testable languages in the strict sense and application to syntactic pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12(9):920–925.

Hayes, G., and Demiris, J. 1994. A robot controller using learning by imitation. In *Proc. of the Intl. Symp. on Intelligent Robotic Systems*, 198–204.

Kaiser, M. 1997. Transfer of elementary skills via human-robot interaction. *Adaptive Behavior* 5(3-4):249–280.

Kuniyoshi, Y.; Inaba, M.; and Inoue, H. 1994. Learning by watching: Extracting reusable task knowledge from visual observation of human performance. *T-RA* 10:799–822.

Nicolescu, M. N., and Matarić, M. J. 2001. Experience-based representation construction: learning from human and robot teachers. In *Proc., IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, 740–745.

Nicolescu, M. N., and Matarić, M. J. 2002. A hierarchical architecture for behavior-based robots. In *Proc., First International Joint Conference on Autonomous Agents and Multi-Agent Systems*.

P. Garcia, E. V., and Casacuberta, F. 1987. Local languages, the successor method and a step towards a general methodology for the inference of regular grammars. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 9(6):841–845.

Schaal, S. 1997. Learning from demonstration. In Mozer, M.; Jordan, M.; and Petsche, T., eds., *Advances in Neural Information Processing Systems 9*, 1040–1046. MIT Press, Cambridge.

Werger, B. B. 2000. Ayllu: Distributed port-arbitrated behavior-based control. In *Proc., The 5th Intl. Symp. on Distributed Autonomous Robotic Systems*, 25–34. Knoxville, TN: Springer.