Learning Prospective Robot Behavior

Shichao Ou and Rod Grupen
Laboratory for Perceptual Robotics
Computer Science Department
University of Massachusetts Amherst
{chao,grupen}@cs.umass.edu

Abstract

This paper presents a learning framework that enables a robot to learn comprehensive policies autonomously from a series of incrementally more challenging tasks designed by a human teacher. Psychologists have shown that human infants rapidly acquire general strategies and then extend that behavior with contingencies for new situations. This strategy allows an infant to quickly acquire new behavior and then to refine it over time. The psychology literature calls such compensatory action *prospective behavior* and it has been identified as an important problem in robotics as well. In this paper, we provide an algorithm for learning prospective behavior to accommodate special-purpose situations that can occur when a general-purpose schema is applied to challenging new cases. The algorithm permits a robot to address complex tasks incrementally while reusing existing behavior as much as possible. First, we motivate prospective behavior in human infants and in common robotic tasks. We introduce an algorithm that searches for places in a schema where compensatory actions can effectively avoid predictable future errors. The algorithm is evaluated on a simple grid-world navigation problem. Results show that learning performance improves significantly over an equivalent flat learning formulation by re-using knowledge as appropriate and extending behavior only when necessary. We conclude with a discussion of where prospective repair of general-purpose behavior can play important roles in the development of behavior for effective human-robot interaction.

Introduction

Human behavior is organized hierarchically and extended over a lifetime of experience with a variety of tasks. This is an open-ended process where the infant extends models and control knowledge incrementally by engaging learning situations near the frontier of his or her abilities. As learning proceeds, the frontier advances into more complex domains and precipitates increasingly expert behavior. This perspective on human development can be successfully applied to robotics as well.

In previous work, we formulated methods for intrinsically motivated learning that creates hierarchical behavior represented as *schema*—general plans for an entire class of tasks (Hart, Sen, and Grupen 2008b; 2008a). We demonstrated that a bimanual robot learns a hierarchy of basic manual skills—searching, grasping, and inspecting objects—by sequencing primitive actions (Hart, Sen, and Grupen 2008b) in search of intrinsic rewards. Schema are acquired initially in a simple learning context devised by the human teacher to make rewards conspicuous. In subsequent stages of development, the robot is challenged with new situations that cause the schema to be extended to make the behavior strictly more comprehensive.

This paper addresses the schema extension process that learns to accommodate new situations where the schema needs to be extended. We propose an algorithm that searches for the state information necessary to recognize the new situation and writes a contingency handler for the new situation using the subgoals that define the schema.

Schema Learning

The use of the term "schema" can be traced back to 1781, where philosopher Immanuel Kant introduced it as a way to map concepts to percepts over categories of objects in order to guard against "thoughts without contents" (Kant 1781). This allowed Kant to talk about mental representation of concepts that are grounded in sensations that would lend support to reasoning and intuition. In the 1950s, Piaget used *schema* to refer to sensorimotor skills that infants use to explore their environments (Piaget 1952). His schema is a mental construction refined through a series of stages by the processes of *assimilation* of new experience and *accommodation* of skills to describe interactions with the environment.

As computational devices, schematic representations have been presented in architectures using planning methods (Lyons 1986), empirical cause-and-effect methods (Drescher 1991), reactive behavior methods (Brooks 1991; Arkin 1998) and rule-based methods (Nilsson 1994). In (Arbib 1995), Robot Schema (RS), a formal language for designing robot controllers has been proposed by Arbib and Lyons, where perceptual and motor schemas are combined into coordinated control programs.

This work is based on a schematic computational framework that takes a control theoretic approach to schema learning (Huber 2000; Hart, Sen, and Grupen 2008b). In this approach, a schema is represented as a collection of sensory and motor resources, and previously learned skills. Through

exploration, the robot discovers which combinations of sensorimotor resources lead to reward. Piaget's notion of *accommodation* and *assimilation* is realized in this framework where existing schemas are factored into *declarative* and *procedural* components, respectively. Declarative structure captures the generalizable sequences of sub-goals that describe the skill and procedural knowledge describes how an existing schematic structure can apply to different run-time contexts. This has been demonstrated in several subsequent stages of development following the acquisition of a basic *search-and-grab* behavior. The separation of declarative and procedural knowledge enabled the robot to quickly adapt to the new situations by preserving the basic *search-and-grab* plan and incorporating handedness, object scale, and shape contingencies and by engaging gestural actions to recruit human assistance. However, the framework does not handle situations where both declarative structure and procedural knowledge of the schema needs to extended simultaneously. In the balance of this paper, a *prospective behavior* algorithm is introduced to address this kind of adaptation.

Prospective Behavior

In general, the repair of a schema in response to a new situation can require a larger temporal scope than indicated solely by the actions that fail. The error can be associated with events that are not monitored by the schema and that occurred at some indefinite time in the past. Prospective behavior is an important component of computational approaches to transfer and generalization. It is a term, coined in the psychology literature, to describe a process in which a human infant learns to predict how a strategy might fail in the future and generates alternative strategies to accommodate the new situation.

McCarty *et al.* studied the initial reach to a spoon laden with applesauce and presented to infants in left and right orientations (McCarty, Clifton, and Collard 1999). The developmental trajectory observed is summarized in Figure 1. Initial policies are biased toward dominant hand strategies that work well when the spoon is oriented with its handle to the dominant side. However, when it is not, the dominant hand strategy fails. Variations in the applesauce reward distinguish important categories in this process—dominant-side and non-dominant-side presentations of the spoon. One hypothesis holds that this process involves a search for perceptual features that distinguish classes of behavioral utility. When this happens, new perceptual features have been learned that were not present in the original representation. They have been selected from a possibly infinite set of alternatives because they form a valuable distinction in the stream of percepts—valued for their ability to increase the reward derived from the infant's interaction with the task.

One may view this process as one in which properties and constraints imposed by the task are incorporated into a policy incrementally starting with the latter (distal) actions and gradually propagating back through the action sequence to early (proximal) actions.

There are parallels to the "pick-and-place" task often studied in robotics (Jones and Lozanzo-Perez 1990). Consider a general purpose pick-and-place schema that acquires an object (the "pick" goal) and delivers it to a desired position and orientation (the "place" goal). A successful grasp of the object can depend on characteristics of the place goal. For instance, if the object is a cylindrical peg that is to be placed at the bottom of a cylindrical hole, then the mating surfaces between the peg and the hole must be left unobstructed for the insertion to succeed. The decision about how to grasp the peg must respect this constraint. Now consider a robot with lots of prior experience with pick-and-place tasks, but none directly focused on the constraints surrounding peg-in-hole insertions. An arbitrary grasp on the peg will likely fail during the place subtask and the reason for this failure is likely inexplicable in the existing pick-and-place framework.

Traditionally, this problem is formulated as a planning problem. In (Lozano-Perez 1981; Jones and Lozanzo-Perez 1990), a back-chaining algorithm is used that searches backward in time from the desired final state until the initial state is found. This approach requires complete knowledge of the task to begin but does not speak to where that knowledge came from. It is subject to uncertainty introduced by seemingly small inaccuracies in backward chaining predictions compounded over multi-step sequences. Moreover, depending on how task knowledge is represented, this strategy may not share common background (pick-and-place) knowledge with other related tasks.

This is in stark contrast to how the human child would approach this problem. Extrapolating from the spoon and applesauce experiment, we expect that the infant will employ a general-purpose strategy and demonstrate biases that apply generally to the entire class of such tasks. Upon failing with this approach, and only upon failing, will the child search for an explanation for the failure, starting at the peg insertion and backing up to the transport phase, to the grasp, and ultimately to the visual inspection of the peg and hole. Somewhere in this sequence is the reason that the general-purpose strategy doesn't work in this context. Once found, the infant will begin experimenting with corrective actions. Throughout this process, the infant's search for a solution revolves around modifying existing behavior rather than attempting to learn a new strategy from scratch.

The work described herein extends our previous work and presents a prospective behavior repair algorithm for autonomous agents to rapidly accommodate a novel task by applying existing behavior. The main idea of the algorithm is the following: upon failure due to a new context, the robot attempts to fix the problem via local adjustments whose scope expands until a compensatory subtask is learned to handle the exception. Now, the general-purpose schema is extended with a call for the compensatory subtask when the triggering percept is present. The result is a new, integrated, and more comprehensive schema that incorporates prospective behavior for accommodating the new context.

In the rest of the paper, we will describe an algorithm for discovering prospective behavior motivated by the behavior of infant learning. Next, we introduce a simple navigation task with multiple "door" contexts that introduce prospective errors. We attempt to show that a general-purpose navigation policy in the grid world can be extended with auxiliary

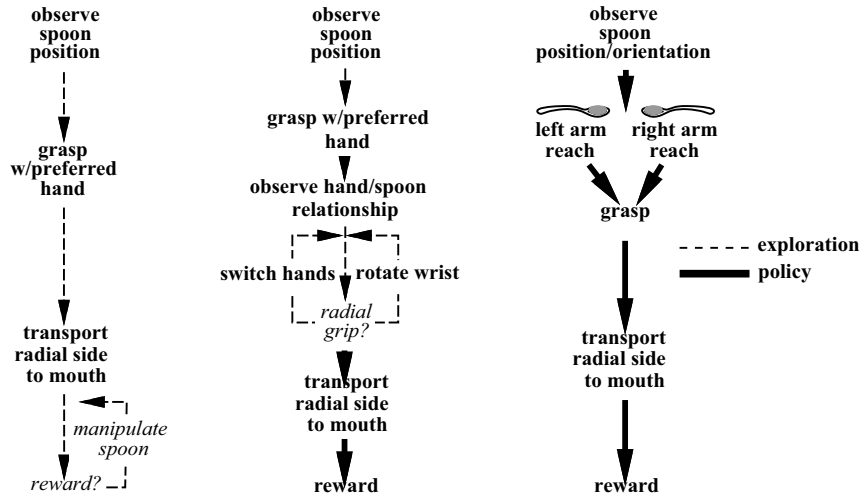


Figure 1: Prospective Behavior revealed in the Applesauce Experiment.

percepts and compensatory actions to solve the problem efficiently. We evaluate the proposed algorithm by comparing its performance to that of a "flat" learning problem in which all the required state information is provided *a priori*.

Related Work

Previous work by Wheeler *et al.* replicated McCarty's applesauce experiment on a bimanual robot platform (Wheeler, Fagg, and Grupen 2002). The robot was first presented with an easier task where the object was always offered in the same orientation. This allowed the robot to quickly learn a dominant hand strategy. Later, the robot was challenged with a more difficult task where the object was presented in random orientations such that if the robot initiated the grasp behavior with the wrong hand, a compensatory strategy was required. Although learning occurred in multiple stages to exhibit a learning progression similar to that reported in the human infant study, Wheeler's learning representation was flat. It did not exploit previously learned skills or sequences of actions for the more challenging tasks.

This work is similar to work by Cohen *et al.* (Cohen, Chang, and Morrison 2007) on hierarchical learning on the aspect that both algorithms autonomously discovers hidden state information that is missing from the current state representation. Cohen uses an entropy approach and we use a decision tree algorithm. Our approach takes another step that actively searches for an appropriate section of the program where the existing policy can be repaired because fixing the problem where it occurs may not yield a solution. Then, a new sub-goal is created such that a prospective behavior can be learned. This aspect of hierarchical learning was not demonstrated in Cohen's work.

Konidaris's work on agent-space options (Konidaris and Barto 2007) studies similar problems in skill transfer where the *agent spaces* become non-Markovian when transfered to new contexts. To resolve the issue, a *problem-space* was introduced that maintains the Markov property. In this work, a similar state factorization technique is employed for a different purpose: to reduce redundant states such that improvement on learning performance can be achieved.

The Navigation Problem

We introduce the prospective repair algorithm by way od a robot navigation task. Figure 2 shows a grid world in which a simulated robot navigates through hallways, rooms, doors, and buttons that actuate the doors. The circle is the robot's starting position and the triangle represents the goal. The robot's task is to learn a path to the goal, given that a random subset of the doors can be closed at the beginning of each training episode. The buttons for opening doors are scattered in different rooms of the map. The robot has to visit the appropriate buttons to open doors that blocks its known path to the goal.

The robot can move left, right, up, or down. At each grid location, the robot can observe its (x, y) location and three door status indicator bits that represent the status of three, randomly chosen doors out of the six in the map. However, the correspondence between the doors and the indicator bits are not directly observable. The initial status of the doors is randomly assigned at the beginning of each trial. We will evaluate two solutions to this problem. The first is a flat learning approach informed by the full state description, and the second is the proposed prospective repair approach using a sequence of reusable policies in (x, y) state with prospective error suppression triggered by the door status indicators.

A Flat Q-learning Approach

A flat learning approach to the problem is formulated where all the required state information is provided *a priori* and the task is presented to the robot in a single learning stage. This is in contrast to the multi-stage learning approach that is presented next. This grid world navigation task is formulated as a standard reinforcement learning problem using the ϵ-greedy Q-learning algorithm (Sutton and Barto 1998) where

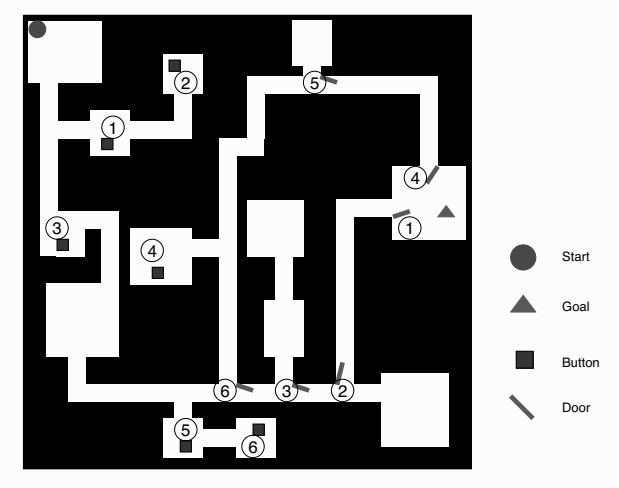


Figure 2: A 30×30 grid-world navigation problem. The status of a door is toggled when the robot visits the grid location where the corresponding button is located.

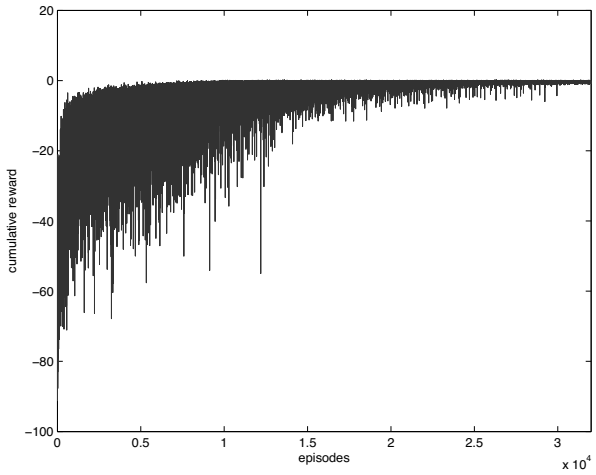


Figure 3: Average cumulative reward over 100 trials for using a flat learning approach

the robot is rewarded for finding an optimal path to the goal. The state, s, for this formulation includes the (x, y) location of the robot and the 3 observable door status indicator bits. The 4 actions: move up, down, left and right, form the robot's the action set $\mathcal{A}$. A simple reward model is applied: the robot receives positive 1 unit of reward for achieving the goal and a -0.01 unit of reward for every step it takes.

In this formulation, the robot receives maximum cumulative reward by taking the fewest number of steps for reaching the goal. For every state s the robot encounters and every action a the robot can take from that state, an expected future reward value, or Q-value is estimated. In the beginning, this value is initialized randomly for every state-action pair $< s, a >$. Through trial-and-error exploration, the Q-learning algorithm enables the robot to incrementally update the Q-value for every $< s, a >$ it encounters. With sufficient exploration, the Q-value for all $< s, a >$ is expected to converge, thus allowing the robot to extract optimal policies for navigating to the goal under all contexts. For these experiments, we define an episode to be one complete traversal by the robot from start position to goal position. Early on, it may take several thousand actions to get to the goal. A trial is defined as one complete learning experiment (until asymptotic performance). Depending on the problem design, it may take consist of several thousand or tens of thousands of episodes before a trial concludes.

The result from the flat learning experiment is presented in Figure 3. In the early episodes, the cumulative rewards are large negative numbers because the robot starts out with no prior knowledge about the world, and randomly explores the map with many extraneous steps, building up large negative reward before finally reaching the goal. Slowly, as expected future reward for each state-action pair improves, the number of steps it takes for the robot to reach the goal decreases. As a result, the cumulative reward rises, until it converges at around $30,000$ episodes. This experiment used a discount factor, $\gamma = 1.0$, learning rate $\alpha = 0.1$, and the ϵ-greedy parameter is set to $\epsilon = 0.1$.

The flat learning approach learns to solve this problem in $30,000$ episodes to learn a policy with contingencies for random door configurations. This is a lot of training for an on-line learner, but further reflection on the experiment yields insights that can be used to reformulate the problem. State s includes the (x, y) location and 3 randomly selected door status bits at each cell in the map. However, in many states, the part of s concerning door status is uninformative and optimal decisions can be determined from (x, y) alone. Therefore, performance in the flat learning problem is often compromised by too much state that is encoded inefficiently. In these states, a more general strategy can be applied and much less training is required. To overcome this problem, the hierarchical prospective repair approach is proposed.

A Prospective Repair Approach

In this section, the proposed prospective repair approach is presented in the context of the multi-door navigation problem. In contrast to the flat-learning approach, the original task is decomposed into a series of problems that can be presented to the robot in an incremental manner. Initially, the robot is presented with the simplest task. Later, it is challenged with more difficult contexts. In the navigation problem, the simplest task is to find the optimal path for reaching the goal when all doors are open. After this policy is acquired, the robot is challenged by closing a specific door until the robot has acquired a policy for handling this case. These skills are reused to construct contingencies for arbitrary door configurations.

The proposed prospective repair algorithm is presented in Algorithm 1. It is divided into 3 main components: (1) a general-purpose strategy is first learned in the simplest context, (2) the robot is challenged with a new context and a auxiliary perceptual feature is learned to differentiate the

new context, and (3) a search is conducted for local repairs whose scope expands until a policy is acquired to handle the exception. Algorithm 1 also depicts the schemas created and/or modified after each of these steps. The proposed approach assumes that a general-purpose strategy exists that applies approximately to the different variations in the task. Subtasks are represented as separate policies to preserve the general-purpose policy to remain unaltered.

As shown in Algorithm 1, human guidance also plays an important role in the prospective repair algorithm, in the form of structured tasks of increasing level of difficulty. The simpler task ensures the robot can quickly learn a basic general-purpose strategy while later tasks allow the robot extend on existing policies and learn to handle more complicated contexts. More importantly, such structured tasks can be created by simple adjustments of environmental constraints at the opportune time of the learning process. For instance, opening or closing doors in the robot navigation domain, or offering correctly oriented spoons in the apple sauce experiments. This form of guidance is intuitive to a human teacher as similar strategies can often be observed in human parent/child interactions (McCarty, Clifton, and Collard 1999).

Multi-stage training sequences provide for behavior reuse, but they are not sufficient for causing an improvement in learning performance. The appropriate state representation and provisions for re-use are required. This is the key difference between this algorithm and previous approaches to prospective behavior using flat learning algorithms(Wheeler, Fagg, and Grupen 2002). The global state of the robot, in this case, is represented using only its (x, y) coordinates. The basic policy relies principally on this information and auxiliary state, i.e. door status indicators, are stored separately and only in places where they are available and needed to trigger contingencies for handling exceptions to the basic plan.

Figure 4 shows the resulting learning curve from the prospective repair/generalization approach applied to the navigation scenario. The action set $\mathcal{A}$ remains the same as in the flat learning formulation. Once again, the robot receives 1 unit of reward for achieving the goal and -0.01 units of reward for every action it takes. The learning parameters, $\gamma = 1.0, \alpha = 0.1$, and $\epsilon = 0.1$ likewise remain the same as in the flat learning problem. In the first stage, a path toward the goal is learned with all the doors open. The initial policy, π, for traversing the unobstructed environment is illustrated in Figure 5). It depends on (x, y) state information exclusively and serves as the initial general-purpose solution. As Figure 4 illustrates, in each subsequent stage, a new context is introduced wherein exactly one of the doors is closed causing the cumulative reward to decline sharply. At this point, a new learning problem is initiated to recognize the new context and to repair the general strategy. Under the experimental conditions described, the reward begins to climb until it converges once again as the robot quickly adapts to the new context. For the particular map used, the closing of some doors do not cause the general policy to fail, therefore there are only 4 dips in the learning curve. The prospective repair process is complete after less than $2,000$ episodes

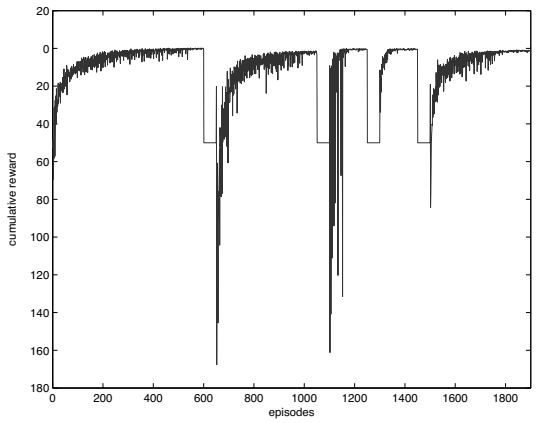


Figure 4: Average cumulative reward over 100 trials using the prospective repair approach. Each dip in the learning curve corresponds to a task change that leads to a specific type of failure in the previously learned policy. Results show that the prospective repair algorithm allows the robot to quickly adapt to each new context.

compared to $30,000$ episodes for the flat-learning approach. We can extrapolate these results and conclude that the advantage would be even more significantly as more doors are added to the map, or when the robot has to pay attention to more perceptual features.

Figure 6 illustrates learned paths to button 1 from any location on the general policy π where the status of the corresponding door can be observed. The path that is the shortest is selected as the compensatory behavior and integrated with the original behavior to achieve a new and more comprehensive behavior.

Several design elements contributed to the performance improvement. First, the choice of the initial state description does indeed provide a policy that serves the task well from many positions in the map—there are only a small number of special cases that the robot must handle. As a result, there is a significantly smaller state-action space than there is with the flat learning approach. All guidance from a human teacher that has this property is expected to produce the same utility in learning performance. Moreover, the search for the prospective behavior is initiated as a separate learning problem with an independent goal and state transition structure, thus enhancing re-use. When multiple doors are closed simultaneously, the prospective repair approach naturally decomposes the original problem into sub-problems associated with navigating to buttons corresponding to closed doors en route to the goal. The robot can reuse previously learned contingencies for relevant doors rather than having to learn them from scratch as in the case of the flat learning design.

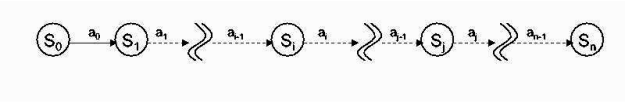
TEACHER

- construct a simple initial training context ⟶ all doors open ⟶

ROBOT

Given a set of percepts: $\mathbf{f} = \{f_1, ..., f_i, f_j, ..., f_n\}$, and actions $\mathcal{A} = \{a_1, ...a_m\}$:

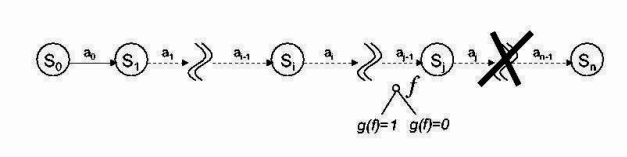
1: Apply factorization technique to define state $s = \{f_1, ..., f_i\}$ where $s \in \mathcal{S}$ contains features that are frequently used for decision making and auxiliary percepts $\mathcal{F} = \{f_j, ..., f_n\}$.
2: Use Q-learning on MDP defined by $< \mathcal{S}, \mathcal{A}, \mathcal{R} >$ to learn a general-purpose policy π, where $\mathcal{R}$ is the predefined reward function for task T.



- challenge the frontier of existing behavior ⟶ close single doors ⟶

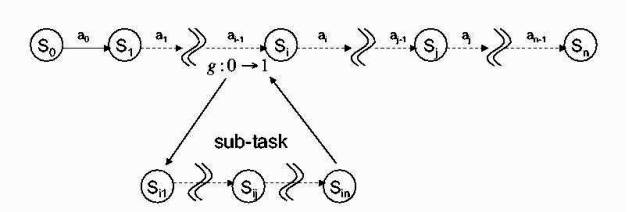
recognize the perceptual associations of the subtask

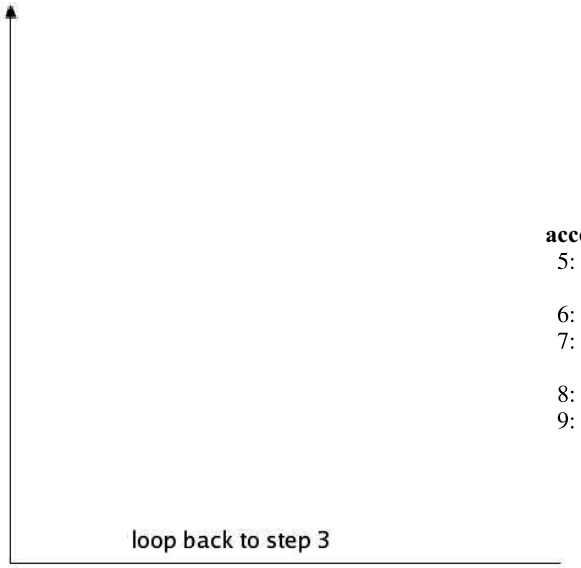
3: Execute policy π until it leads to repeated failure and accumulate experience data set, $\mathcal{D}$, recording features $\mathbf{f} \in \mathcal{F}$ and the success or failure of π in that context.
4: Apply a generic discriminative learning algorithm (e.g. $C4.5$) on $\mathcal{D}$ to identify a decision boundary $g(\mathbf{f})$ that differentiates success and failure under policy π. Function g is said to *accept* $\mathbf{f}$ if it predicts success under policy π.



accommodate the new context

5: Create a new MDP defined by $< \mathcal{S}, \mathcal{A}, \mathcal{R}' >$, where $\mathcal{R}'$ is a reward for restoring $\mathbf{f}$ to the condition where g accepts $\mathbf{f}$.
6: **for all** states $s \in \mathcal{S}$ in which g does not accept $\mathbf{f}$ **do**
7: Starting from s, learn a compensatory policy π_g for achieving the sub-goal defined by g.
8: **end for**
9: Merge π_g with π to form a new hybrid policy π'.





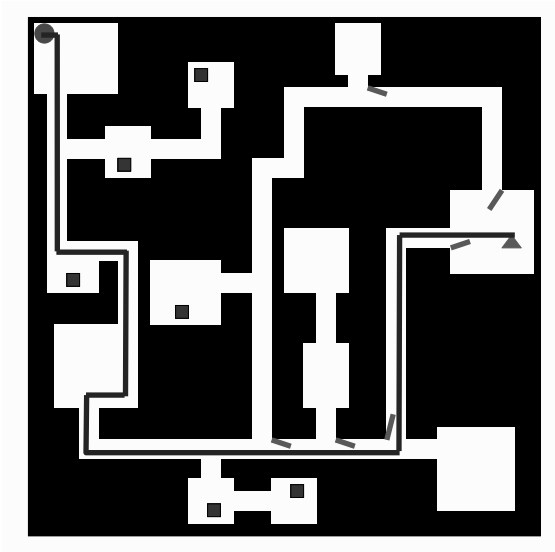


Figure 5: Learning result from stage 1: an unobstructed path π to the goal that functions as the general-purpose policy.

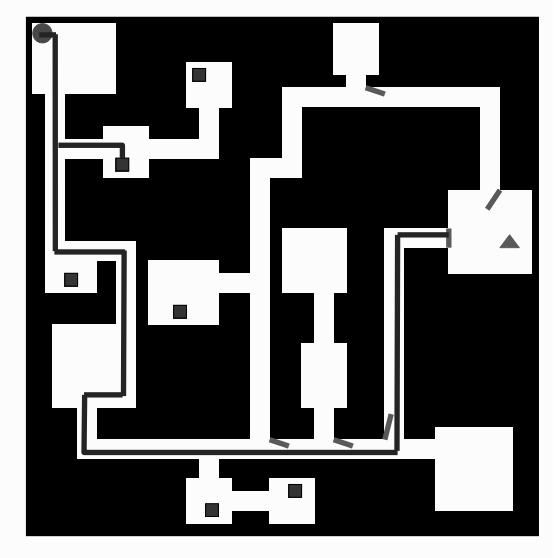


Figure 6: Learned paths to the button 1 for opening door 1 from any location on the general policy π where the status of the corresponding door can be observed. By integrating this policy with π, a new, more comprehensive policy for handling the contingency of the closing of door 1 can be created.

Conclusion and Discussion

This work advocates an incremental learning paradigm towards behavior acquisition in robots, where a human user can teach robots skills interactively, using a sequence of increasingly challenging tasks. This is an open-ended process that requires learning framework designers to build systems that can act based on incomplete information and that adapt to new situations where previously learned behavior fails.

In this work, human guidance first comes in the form of training guidance—structuring the environment and focusing exploration on a restricted set of sensors and effectors and thus states and actions in order to facilitate the formation of new skills. In subsequent stages, constraints are incrementally removed.

The proposed prospective repair algorithm has significant learning performance advantage over the flat Q-learning approach for solving tasks that can be decomposed into a series of problems and presented to the robot in an incremental fashion. The significant improvement is the result of knowledge reuse including maintaining much of the previously learned path in the new strategy, and only learn a new compensatory policy such that doors blocking the path to the goal can be re-opened. Once the robot has learned how to open any door individually, this knowledge is reused again for the case where multiple doors are closed simultaneously, thus minimizing redundant learning.

This paper offers a developmental view of learning and teaching robot skills and makes a case for how this can be achieved using the proposed learning framework to enable a robot learn and refine skills incrementally through structured learning stages provided by a human teacher.

Acknowledgments

This research is supported under the NASA-STTR-NNX08CD41P, ARO-W911NF-05-1-0396, and ONR-5710002229. The authors would also like to acknowledge Stephen Hart and Shiraj Sen for their helpful discussions.

References

Arbib, M. 1995. Schema theory. In *The Handbook of Brain Theory and Neural Computation*, 830–834. Cambridge, MA: MIT Press.

Arkin, R. C. 1998. *Behavior-Based Robotics*. MIT Press.

Bernstein, D. S. 1999. Reusing old policies to accelerate learning on new MDPs.

Brooks, R. 1991. Intelligence without representation. *Artificial Intelligence Journal* 47:139–159.

Cohen, P.; Chang, Y. H.; and Morrison, C. T. 2007. Learning and transfering action schemas. In *Proceedings of IJCAI*.

Drescher, G. 1991. *Made-Up Minds: A Constructionist Approach to Artificial Intelligence*. Cambridge, MA: MIT Press.

Hart, S.; Sen, S.; and Grupen, R. 2008a. Generalization and transfer in robot control. In *Epigenetic Robotics Annual Conference*.

Hart, S.; Sen, S.; and Grupen, R. 2008b. Intrinsically motivated hierarchical manipulation. In *Proceedings of 2008 IEEE Conference on Robots and Automation (ICRA)*.

Huber, M. 2000. *A Hybrid Architecture for Adaptive Robot Control*. Ph.D. Dissertation, Department of Computer Science, University of Massachusetts Amherst.

Jones, J., and Lozanzo-Perez, T. 1990. Planning two-fingered grasps for pick-and-place operations on plyhedra. In *Proceedings of 1990 Conference on Robotics and Automation*.

Kant, I. 1781. *Critique of Pure Reason, Translated by Norman Kemp Smith*. Macmillan & Company, Ltd.

Konidaris, G., and Barto, A. 2007. Building portable options: Skill transfer in reinforcement learning. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, 895–900.

Lozano-Perez, T. 1981. Automatic planning of manipulator transfer movements. In *Trans. Syst. Man, Cybern.*, volume SMC-11, 681–698.

Lyons, D. 1986. RS: A formal model of distributed computation for sensory-based robot control. Technical Report 86-43, COINS Computer Science, University of Massachusetts, Amherst.

McCarty, M.; Clifton, R.; and Collard, R. 1999. Problem solving in infancy: The emergence of an action plan. *Developmental Psychology* 35(4):1091–1101.

Nilsson, N. 1994. Teleo-reactive programs for agent control. *Journal of Artificial Intelligence Research* 139–158.

Piaget, J. 1952. *The Origins of Intel ligence in Childhood*. International Universities Press.

Simsek, Ö.; Wolfe, A. P.; and Barto, A. G. 2005. Identifying useful subgoals in reinforcement learning by local graph partitioning. In Raedt, L. D., and Wrobel, S., eds., *ICML*, volume 119 of *ACM International Conference Proceeding Series*, 816–823. ACM.

Sutton, R., and Barto, A. 1998. *Reinforcement Learning*. Cambridge, Massachusetts: MIT Press.

Thrun, S., and Schwartz, A. 1995. Finding structure in reinforcement learning. In Tesauro, G.; Touretzky, D.; and Leen, T., eds., *Advances in Neural Information Processing Systems*, volume 7, 385–392. The MIT Press.

Vygotsky, L. 1930. *Mind in society*. Harvard University Press.

Wheeler, D.; Fagg, A.; and Grupen, R. 2002. Learning prospective pick and place behavior. In *Proceedings of the IEEE/RSJ International Conference on Development and Learning*.