

2006

Socially Guided Machine Learning

Andrea Lockerd Thomaz
Massachusetts Institute of Technology

Cynthia Breazeal
Massachusetts Institute of Technology

Andrew G. Barto
University of Massachusetts - Amherst

Rosalind Picard
Massachusetts Institute of Technology

Follow this and additional works at: https://scholarworks.umass.edu/cs_faculty_pubs



Part of the [Computer Sciences Commons](#)

Recommended Citation

Thomaz, Andrea Lockerd; Breazeal, Cynthia; Barto, Andrew G.; and Picard, Rosalind, "Socially Guided Machine Learning" (2006).
Computer Science Department Faculty Publication Series. 183.
Retrieved from https://scholarworks.umass.edu/cs_faculty_pubs/183

This Article is brought to you for free and open access by the Computer Science at ScholarWorks@UMass Amherst. It has been accepted for inclusion in Computer Science Department Faculty Publication Series by an authorized administrator of ScholarWorks@UMass Amherst. For more information, please contact scholarworks@library.umass.edu.

Socially Guided Machine Learning

by

Andrea Lockerd Thomaz

B.S., University of Texas, Austin (1999)

S.M., Massachusetts Institute of Technology (2002)

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Media Arts and Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2006

© Massachusetts Institute of Technology 2006. All rights reserved.

Author
Program in Media Arts and Sciences
May 5, 2006

Certified by
Cynthia Breazeal
Associate Professor of Media Arts & Sciences
Thesis Supervisor

Accepted by
Andrew Lippman
Chairman, Department Committee on Graduate Students

Socially Guided Machine Learning

by

Andrea Lockerd Thomaz

B.S., University of Texas, Austin (1999)

S.M., Massachusetts Institute of Technology (2002)

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Media Arts and Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2006

© Massachusetts Institute of Technology 2006. All rights reserved.

Certified by.....

Andrew G. Barto
Professor of Computer Science
University of Massachusetts, Amherst
Thesis Reader

Socially Guided Machine Learning

by

Andrea Lockerd Thomaz

B.S., University of Texas, Austin (1999)

S.M., Massachusetts Institute of Technology (2002)

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Media Arts and Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2006

© Massachusetts Institute of Technology 2006. All rights reserved.

Certified by.....

Rosalind Picard

Professor of Media Arts & Sciences

Massachusetts Institute of Technology

Thesis Reader

Socially Guided Machine Learning

by

Andrea Lockerd Thomaz

Submitted to the Program in Media Arts and Sciences
on May 5, 2006, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Media Arts and Sciences

Abstract

Social interaction will be key to enabling robots and machines in general to learn new tasks from ordinary people (not experts in robotics or machine learning). Everyday people who need to teach their machines new things will find it natural for to rely on their interpersonal interaction skills. This thesis provides several contributions towards the understanding of this Socially Guided Machine Learning scenario.

While the topic of human input to machine learning algorithms has been explored to some extent, prior works have not gone far enough to understand what people will try to communicate when teaching a machine and how algorithms and learning systems can be modified to better accommodate a human partner. Interface techniques have been based on intuition and assumptions rather than grounded in human behavior, and often techniques are not demonstrated or evaluated with everyday people.

Using a computer game, *Sophie's Kitchen*, an experiment with human subjects provides several insights about how people approach the task of teaching a machine. In particular, people want to direct and guide an agent's exploration process, they quickly use the behavior of the agent to infer a mental model of the learning process, and they utilize positive and negative feedback in asymmetric ways. Using a robotic platform, Leonardo, and 200 people in follow-up studies of modified versions of the *Sophie's Kitchen* game, four research themes are developed.

The use of human *guidance* in a machine learning exploration can be successfully incorporated to improve learning performance. Novel learning approaches demonstrate aspects of *goal-oriented learning*. The *transparency* of the machine learner can have significant effects on the nature of the instruction received from the human teacher, which in turn positively impacts the learning process. Utilizing *asymmetric* interpretations of positive and negative feedback from a human partner, can result in a more efficient and robust learning experience.

Thesis Supervisor: Cynthia Breazeal

Title: Associate Professor of Media Arts & Sciences

Acknowledgments

As my thesis is about the social structure of learning environments, it seems very natural to start by recognizing the amazing and supportive environment that I have found myself in over the past several years.

First, Cynthia Breazeal, for being a great advisor, and for her contagious enthusiasm and intensity. Her work and her vision of the future are truly ground breaking and I am grateful to her for not only having given me the opportunity to be a part of it, but having wholeheartedly supported my blossoming research agenda every step of the way.

In my years at the Media Lab, Roz Picard has been a sincerely positive presence, as a researcher, a teacher and a person. Her excitement and curiosity in her research questions, and her priorities in work and life are an inspiration and a model.

It is great to have had the opportunity to work with Andy Barto. His open-minded view of the Machine Learning field has made for some inspiring discussions, and his support of a newcomer with a different perspective is very encouraging.

The Robotic Life Group has been a great environment and a wonderful group of people to work with over the last few years, especially everyone working on Leonardo. Special thanks to Jeff Lieberman and Dan Stiehl who have, at critical moments, given their time freely; Leo thanks you! To everyone in the Leo behavior system ranks: Matt Berlin, Zoz Brooks, Jesse Gray, and Guy Hoffman. It's truly a pleasure to work with people who really love and believe in what they do. It's been a lot of work, but a whole lot of fun to work with you all.

The Media Lab has been an incredible environment, full of excitement and opportunity. I would like to thank my first advisor, Ted Selker, for introducing me to the lab and continuing to be a positive and creative influence. Thanks also to all the faculty who have had an impact on my research in direct and indirect ways: Henry Lieberman, Patrick Winston, John Maeda, Bruce Blumberg, Pattie Maes, Deb Roy, Glorianna Davenport. Thanks in particular to Mitch Resnik for being on my general exams committee and helping to introduce me to the field of Situated Learning.

Thanks especially to my parents and my brothers, who have always believed in me, and their encouragement and support has shaped the person that I am today in countless ways. And a special thanks to my wonderful husband, Edison Thomaz. I could not have made it through all of the hard work in the last few years without his love and support.

Contents

1	Introduction	23
1.1	Motivation	24
1.1.1	Learning is a part of all activity	25
1.1.2	Teachers scaffold the learning process	25
1.1.3	Expression provides feedback to guide a teacher	26
1.2	Machine Learning Background	27
1.2.1	Supervised Learning	27
1.2.2	Unsupervised Learning	28
1.2.3	Semi-Supervised Learning	28
1.2.4	Reinforcement Learning	28
1.2.5	The role of the human in standard ML approaches	29
1.3	Related Work	30
1.3.1	Approaches designed for human input	30
1.3.2	An Interaction perspective of ML	33
1.4	Thesis Overview	35
2	Experiments in Socially Guided Machine Learning	37
2.1	The <i>Sophie's Kitchen</i> Platform	38
2.1.1	Sophie's MDP	38
2.1.2	Learning Algorithm	40
2.1.3	Interactive Rewards Interface	41
2.2	Experimental Design	41
2.3	Findings	42
2.3.1	Guidance Intentions	42
2.3.2	Inferring a Model of the Learner	44
2.3.3	An Asymmetric Use of Rewards	46
3	Utilizing Social Guidance	47
3.1	The Leonardo Robot Platform	48

3.1.1	Sensory Inputs	48
3.1.2	Cognitive Architecture	49
3.2	A Socially Guided Learning Dialog	50
3.2.1	Task Representation	52
3.2.2	Learning Mechanism	54
3.2.3	Hypothesis Expansion and Generalization	55
3.2.4	Execution of a Known Task	57
3.2.5	Example Learning Results	58
3.3	Using Guidance in Sophie's Kitchen	60
3.3.1	Modifications to Leverage Human Guidance	60
3.3.2	Evaluation	62
3.4	Socially Guided Exploration	64
3.4.1	Foundations for Self-Motivated Exploration	65
3.4.2	Action System Overview	68
3.4.3	Task Learning Action Group	69
3.4.4	Task Representation	72
3.4.5	Learning Task Option Policies	74
3.4.6	Task Generalization	76
3.4.7	Scaffolded Learning	80
3.4.8	Example learning results	81
3.5	Human Guidance for Machine Learning Systems	88
4	Transparency to Guide a Human Teacher	91
4.1	Effects of Transparency in <i>Sophie's Kitchen</i>	92
4.1.1	Sophie's Gazing Behavior	92
4.1.2	Experimental Design	93
4.1.3	Result: Gaze Improves Guidance	94
4.2	Nonverbal Transparency Devices on Leonardo	95
4.2.1	Social cues for Scaffolding	95
4.2.2	Facial Expressions to Reveal Internal Learning State	97
4.3	Effects of Leonardo's Nonverbal Communication	99
4.3.1	Experiment	99
4.3.2	Procedure	101
4.3.3	Results	101
4.4	Transparent Learning Machines	104

5	The Asymmetry of Human Feedback	107
5.1	Negative Feedback Leading to Refinement	107
5.1.1	Task Execution and Refinement	108
5.1.2	Just-in-Time Correction	108
5.2	Negative Feedback Leading to Action Reversal	109
5.2.1	Modification for Sophie's UNDO Response	110
5.2.2	Evaluation	110
5.3	Asymmetric use of Feedback in Machine Learning	112
6	Contributions	115
6.1	Experimental findings about how people want to teach	116
6.2	The Guidance-Exploration Spectrum	116
6.3	Guidance with Everyday Human Trainers	117
6.4	Transparency to Improve the Learning Environment	117
6.5	Asymmetric Interpretations of Human Feedback	118
6.6	Mechanisms of Goal-oriented Learning	119
6.7	Concluding Remarks	120
A	Sophie's Kitchen Experiments	123
A.1	Experiment 1 – in Lab	123
A.1.1	Experimental Protocol	123
A.1.2	Informed consent signed by each participant	124
A.1.3	Written instructions given to participants	126
A.1.4	Questionnaire Completed by participants	128
A.1.5	Informal Interview	129
A.2	Experiment 2 – Online	129
A.2.1	Experimental Protocol	129
A.2.2	Introduction page	130
A.2.3	Informed consent page	131
A.2.4	Instructions	131
A.3	Guidance Experiment – in Lab	132
B	Sphinx Grammar	133
B.1	Full JSGF Grammar with Parse Tags	133

List of Figures

1-1	SG-ML explicitly acknowledges the human in the loop, in contrast to standard supervised ML techniques.	33
2-1	<i>Sophie's Kitchen</i> . The agent is in the center, with a shelf on the right, oven on the left, a table in between, and five cake baking objects. The vertical bar is the interactive reward and is controlled by the human.	39
2-2	There is one mark for each player, indicating their percentage of object rewards that were about the last object of attention. This graph shows that many people had object rewards that were rarely about the last object, thus rarely used in a feedback orientation.	43
2-3	A reward to the empty bowl or tray on the shelf is assumed to be meant as guidance instead of feedback. This graph shows that 15 of the 18 players gave rewards to the bowl/tray empty on the shelf.	43
2-4	Ratio of rewards to actions over the first three quarters of the training sessions shows an increasing trend.	44
2-5	Each bar represents an individual and the height is the percentage of object rewards. The difference in the first and last training quarters shows a drop off in usage over time.	45
2-6	Histograms of rewards for each individual in the first quarter of their session. The left column is negative rewards and the right is positive rewards. Most people even in the first quarter of training have a much higher bar on the right.	46
3-1	Leo and his workspace with three button toys.	48
3-2	An overview of the states and flow of execution in the Task Learning Module, which allows Leo to learn from a human partner within a social dialog.	51
3-3	The hypothesis space of goal beliefs expanded from the common goal belief x_{CGB} with two expectation features $\{Y, Z\}$, and four criteria features $\{A, B, C, D\}$	56

3-4	Learning to turn two buttons ON and OFF, and the progressive task and goal representation. Initially there are two buttons in front of Leo, Button1 and Button2, and they are both in the OFF state.	58
3-5	Four trials of an interaction in which a human (H) teaches Leo (L) to “Turn the buttons ON.” From left to right the buttons are red, green, and blue. An ON button is indicated with a star, OFF does not have the star.	59
3-6	The embellished communication channel includes the feedback messages as well as guidance messages. In 3-6(a), feedback is given by left-clicking and dragging the mouse up to make a green box (positive) and down for red (negative). In 3-6(b), guidance is given by right-clicking on an object of attention, selecting it with the yellow square.	61
3-7	Each of Leo’s motivational drives has an initial value and a specified range. Within this range it has a set point (the value that it drifts towards).	67
3-8	Leonardo’s Action System has several Actions and Action Groups that compete for control of the behavior at any given time. For the purpose of this thesis the primary focus is the Task Learning Action Group. This group becomes relevant (triggers) in several learning contexts and utilizes various specific actions in these contexts, described in Sec. 3.4.3	68
3-9	The Task Learning Action Group has three competing actions, this figure shows the nine learning contexts in which each action is available.	71
3-10	The logic executed when each of the three learning actions is triggered.	72
3-11	Between-policy generalization example: Fig. 3-11(a) shows the generalization for the example where the two tasks have similar goals and action policies. Fig. 3-11(b) shows the example where they have similar goals but different action policies.	79
3-12	Leo’s playroom, experimental scenarios for Guided Exploration in both the virtual and physical world.	82
3-13	Guided Exploration learning example: Leo learns about opening two different kinds of boxes. He is able to generalize about flipping a switch ON (T1, T3, T4, T5, and T6), he learns to open each one (T1, T7) and between-policy generalization makes a general task about opening with the specific policies, within-policy generalization simplifies it further (T8). Due to space, some of the intermediate tasks are not pictured.	84

3-14	A snapshot of approximately 10 minutes of a learning session. The top graph shows the dynamics of the motivational drives and the bottom graph shows the resulting dynamics of the learning behaviors. This segment starts with a period where more Relevance actions are being triggered, and mastery starts to rise. This is followed by a period of exploration interspersed with learning about novel states, and then more practicing is seen.	86
3-15	An experimental learning session in the virtual playroom. The graph shows how the size of the set <i>Tasks</i> grows and changes over time. In ‘OrigTasks’ series of data shows the number of $T \in Tasks$ that exist in their original form as created by the novelty action. In the ‘GenTasks’ series we see the number of $T \in Tasks$ over time that are a generalized version. Initially, the OrigTasks number increases as new tasks are learned, and as generalization begins to happen, GenTasks increases and OrigTasks number decreases. Then halfway through the training session, when a new object is introduced, a number of new tasks are created so OrigTasks increases again, but then decreases as these also become generalized with experience. After a 25 minute training session, very few $T \in Tasks$ are in their original formulation, they have been refined and generalized through experience and practice.	87
4-1	Two figures illustrating Sophie’s gazing transparency behavior. In Fig. 4-1(a) Sophie is facing the shelf, gazing at the tray prior to selecting a next action; in Fig. 4-1(a) at the bowl.	92
4-2	The extreme poses representing the extent of Leo’s emotional facial expression used for transparency in motivated learning with guided exploration.	97
4-3	Leo and his workspace with three buttons and a human partner.	100

List of Tables

3.1	An expert user trained 20 agents, with and without guidance, following a strict best-case protocol in each condition; this yields theoretical best-case effects of guidance on learning performance. (F = failed trials, G = first success). The following are the results of 1-tailed t-tests.	63
3.2	Non-expert human players trained Sophie with and without guidance communication available and also show positive effects of guidance on the learning performance. (F = failed trials, G = first success). The following are the results of 1-tailed t-tests.	64
4.1	1-tailed t-test showing the effect of gaze on guidance. Compared to the guidance distribution without gaze, the gaze condition caused a decrease when uncertainty was low and an increase when uncertainty was high. (uncertainty low = number of action choices ≤ 3 , high = number of choices ≥ 3).	94
4.2	Social Cues for Scaffolding	96
4.3	This table is a summary of a table from [Smith and Scott, 1997], showing the various proposed meanings (pleasantness, goal obstacle/discrepancy, anticipated effort, attentional activity, certainty, novelty, personal agency/control) of several individual facial action units. (+) indicates that the facial action is hypothesized to increase with increasing levels of the meaning; (-) indicates that the facial action is hypothesized to increase with decreasing levels of the meaning. These meanings inspire the facial expressions chosen to act as transparency devices in Leo's Guided Exploration.	98
4.4	Leonardo's Facial Expressions to Reveal Learning State in the Guided Exploration implementation.	99
4.5	Time to complete the overall task as a function of the number of errors (e).	102
4.6	Time to complete the labeling portion of the task for each case as a function of the number of errors (e).	103

5.1	1-tailed t-test: Significant differences were found between the baseline and undo conditions, in training sessions with nearly 100 non-expert human subjects playing the <i>Sophie's Kitchen</i> game online.	112
-----	---	-----

List of Algorithms

1	Q-Learning with Interactive Rewards from a Human Partner	40
2	Interactive Q-Learning modified to incorporate interactive human guidance in addition to feedback.	61
3	With each experience $(s_1, a \rightarrow s_2)$, every task has the opportunity to learn, with the possibility of both extending and updating its policy.	75
4	Interactive Q-Learning with guidance and a gazing transparency behavior.	93
5	Interactive Q-Learning with the addition of the UNDO behavior	109

Chapter 1

Introduction

The use of robots in everyday human environments has long been a goal of scientists and a vision of novelists and screenwriters (picture R2D2 of Star Wars, or Rosie of The Jetsons). This vision alludes to robots that are able to communicate, cooperate, collaborate, and coexist with their human partners. Several realms of academia and industry are actively at work toward this goal. For example, putting robots into homes to assist the elderly, or into space as cooperative partners for astronauts. However, a key problem remains unsolved and relatively unexplored: social learning will be crucial to the successful application of robots in everyday human environments. It will be impossible to give these machines all of the knowledge and skills a priori that they will need to serve useful long term roles in our dynamic world. The ability for naïve users, not experts, to guide them easily will be key to their success. While recognizing the success of current machine learning techniques over the years, these techniques have not been designed for learning from non-expert users and are generally not suited for it ‘out of the box’.

The cornerstone of this research is the belief that machines designed to interact with people to learn new things should utilize behaviors and conventions that are socially relevant to the humans with which they interact. They should more fully be able to participate in the teaching and learning partnership, a two-way collaboration. Moreover, the ability to utilize and leverage these social skills is more than a good interface for people, it can positively impact the underlying learning mechanisms to let the system succeed in a real-time interactive learning session.

This thesis concerns *Socially Guided Machine Learning* (SG-ML), exploring the ways in which machine learning can exploit social learning. First, three dimensions of SG-ML are highlighted in a study with human subjects: Guidance, Transparency, and Asymmetry. Then each of these dimensions are explored through software and robotic implementations and experiments. This work demonstrates explicit performance benefits of incorporating social interaction into the machine learning process.

1.1 Motivation

This research is motivated by the distinction between human learning and machine learning. In aiming to build more flexible, efficient, personable and teachable machines, child development and the human learning process serve as inspiration and direction. Children naturally interact with adults and peers to learn new things in social situations. Children are motivated learners that seek out and recognize learning partners and learning opportunities. Additionally, throughout their development, children’s learning is aided in crucial ways by the structure and support of their environment and especially their social environment. A primary hypothesis of this work is that a machine will learn better from humans if it is given the ability to take advantage of the social structure provided by interacting with a human partner or teacher.

Situated learning is a field of study that looks at the social world of a child and how it contributes to their development. One key concept is ‘scaffolding’, where an adult organizes a new skill into manageable steps and provides support such that a child can achieve something they would not be able to accomplish independently [L. S. Vygotsky, 1978, Greenfield, 1984].

In a situated learning interaction, a good instructor maintains a mental model of the learner’s understanding and structures the learning task appropriately with timely feedback and guidance. The learner contributes to the process by expressing their internal state via communicative acts (e.g., expressing understanding, confusion, attention, etc.). This reciprocal and tightly coupled interaction enables the learner to leverage from instruction to build the appropriate representations and associations.

When a machine learner can assume that learning is taking place in the presence of a human that is motivated to help, social interaction can be a key element in the success of the learning process, constraining and assisting the machine. A good teacher will scale instruction appropriately and create a good environment for learning the task at hand. In particular the human may be able to help the robot with hard problems like: “what to learn,” “when to learn,” “what action to try,” and “how to measure success” [Breazeal, 2002].

This situated learning process stands in contrast to typical scenarios of machine learning which are often not interactive nor intuitive for the human partner. With the belief that the human can provide more than labeled examples or a reinforcement signal, this research focuses on three key qualities that distinguish natural learning systems from machine learning systems: motivation, scaffolding, and expression. This section highlights evidence from human tutelage and child development around these topics.

1.1.1 Learning is a part of all activity

In most machine learning examples, learning is an explicit activity. The system is designed to learn a particular thing at a particular time. With humans on the other hand, there is a motivation for learning, a drive to be a better “system”, and an ability to seek out the expertise of others. Some characteristics of a motivated learner include:

- The ability to recognize and exploit good sources of information
- The ability to adopt such an information source as a role model, and a desire to ‘be more like’ that role model that underlies all activity.
- The ability to judge one’s success at an attempted skill, and to have both success and failure experiences affect one’s motivation level in an appropriate way.
- A curiosity about new environments and experiences.
- A sense of one’s level of mastery with acquired skills driving motivation to explore and learn about the world at opportune times.

Learning is not activity, but is part of all activity. This is central to Lave and Wenger’s theory of ‘Legitimate Peripheral Participation’, highlighting that learning is motivated by a learner’s desire to form their identity and become a full participant in the world [Lave and Wenger, 1991].

Children put themselves in a good position to learn new things by being able to recognize and seek proximity to their caregivers. They assume that the caregiver has their best interest in mind and even very young infants use this to their advantage when faced with an unknown situation [Rogoff and Gardner, 1984]. A critical part of learning is gaining the ability to exploit the expertise of others [Pea, 1993].

The ability and desire to engage, communicate, and interact with others is seen from an early age. By the time infants are two months old, they can actively engage in communicative interactions or turn-taking routines with adults. Studies have shown that infants can start and stop communication with their mother through gesture and gaze, and that it is the infants that control the pace of the turn taking interaction [Trevvarthen, 1979, Kaye, 1977]. This turn taking capability is the foundation of many situated learning activities, and is a precursor to more sophisticated interactions like imitation and scaffolding [Zukow-Goldring et al., 2002, Greenfield, 1984].

1.1.2 Teachers scaffold the learning process

An important characteristic of a good learner is the ability to learn both on one’s own and by interacting with another. Children are capable of exploring and learning on their

own, but in the presence of a teacher they can take advantage of the social cues and communicative acts provided to accomplish more. For instance, the teacher often guides the child's search process by providing timely feedback, luring the child to perform desired behaviors, and controlling the environment so the appropriate cues are easy to attend to, thereby allowing the child to learn more effectively, appropriately, and flexibly.

Attention direction is one of the essential mechanisms that contributes to the learning process [Wertsch et al., 1984, Zukow-Goldring et al., 2002]. Analyzing parent-child tutoring sessions reveals a number of ways that adults provide structure and guide attention to let children succeed: placing important objects close to the child's face, arranging the physical environment such that the desired action is within reach, or doing a demonstration in the infant's line of sight to introduce object affordances. The adult is also implicitly directing the child's attention with their gaze direction.

Dynamic Scaffolding is the notion that adults create a learning situation that is the right level of complexity for the learner. The adult adjusts dynamically to make sure the child is working within the Zone of Proximal Development. One way to describe this is that the teacher creates 'microworlds' for the learner to master parts of the task in isolation before moving on, providing safety and intermediate attainable goals [Burton et al., 1984]. For example, with language parents first treat anything as conversational speech, but eventually they raise their expectations, scaffolding the child's conversational abilities [Trevvarthen, 1979].

Linking New and Old: An important role that the adult plays in a child's learning process is linking new information to old, showing or suggesting to the child similarities between new problems and old ones [Rogoff and Gardner, 1984]. A good teacher makes the information in a new problem compatible with what is known, guiding the generalization process, helping the child apply skills across various contexts.

1.1.3 Expression provides feedback to guide a teacher

To be a good instructor, one must maintain a mental model of the learner's state (e.g., what is understood so far, what remains confusing or unknown) in order to appropriately structure the learning task with timely feedback and guidance. The learner helps the instructor by expressing their internal state via communicative acts (e.g., expressions, gestures, or vocalizations that reveal understanding, confusion, attention, etc.). Through reciprocal and tightly coupled interaction, the learner and instructor cooperate to help both the instructor to maintain a good mental model of the learner, and the learner to leverage from instruction to build the appropriate models, representations, and associations.

This human-style tutelage is a social and fundamentally cooperative activity. There-

fore theories of human cooperative and collaborative activity help inform the design of SG-ML systems. These theories argue for the importance of sharing information through communication.

Cohen et al. analyzed task dialogs, where an expert instructs a novice assembling a physical device, and found that much of task dialog can be viewed in terms of joint intentions. Their study identified key discourse functions including: organizational markers that synchronize the start of new joint actions ("now," "next," etc.), elaborations and clarifications for when the expert believes the apprentice does not understand, and confirmations establishing the mutual belief that a step was accomplished [Cohen et al., 1990].

Bratman defines prerequisites for an activity to be considered shared and cooperative: he stresses the importance of mutual responsiveness, commitment to the joint activity and commitment to mutual support [Bratman, 1992]. Cohen et al. support these guidelines and also predict that an efficient and robust collaboration scheme in a changing environment needs an open channel of *communication*.

An SG-ML system that people will find collaborative and cooperative, must take into account nonverbal communication (like gesture [Krauss et al., 1996] and gaze [Argyle et al., 1973]) to facilitate the interaction and maintain an understandable transparent interface between the human and the machine.

1.2 Machine Learning Background

Much of Machine Learning (ML) can be characterized as discovering the structure that is in some data or in the world through sophisticated statistical learning techniques. This section gives a very brief overview of the areas of ML theory discussed throughout this thesis.

1.2.1 Supervised Learning

Supervised learning systems typically learn a mapping between input and output through statistical analysis of hundreds or thousands of training examples chosen by a 'knowledgeable supervisor'. Each example contains both the input features and the desired output value or label (for greater detail see [Duda et al., 2002]). These techniques rely on the availability of labeled data, and are not appropriate in domains with a small number of examples. They are also not appropriate when the environment is changing so quickly that earlier examples are no longer relevant.

1.2.2 Unsupervised Learning

Unsupervised systems learn using only the input set, without output labels (for an introduction see [Duda et al., 2002]). A common approach is clustering, where given some means of comparing the various features of the data (distance metrics) the system can find subsets or clusters of the training examples that are similar. Other approaches try to fit the data set to a model, e.g., a Bayesian approach treats inputs as latent variables and builds a joint density model for the data set. The success of unsupervised approaches again relies heavily on the availability of a large amount of training data.

1.2.3 Semi-Supervised Learning

Semi-supervised learning is a relatively recent area of research that combines unsupervised and supervised learning approaches. Generally these approaches use unsupervised learning techniques to learn the structure of the data, making it easier to identify the ‘most interesting’ examples in a training set. This can then bootstrap a supervised learning technique gaining better performance with fewer labeled examples. For example, active learning is one such approach [Cohn et al., 1995].

1.2.4 Reinforcement Learning

Reinforcement Learning (RL) is commonly used for systems that need to learn from self-generated experience over time – for an introduction see [Sutton and Barto, 1998]. A widely known RL algorithm is *Q-Learning* [Watkins and Dayan, 1992]. In Q-Learning, it is assumed that the agent can perceive the environment as being in one of a finite number of *states*. A state can be thought of as a feature vector from the agent’s sensory input devices (which can be both internal and external aspects of the environment). From any state there are a finite number of *actions* that the agent can execute. It is assumed that at any time the agent will select only one action which may or may not transition the agent from the current state into a new state of the environment. The agent receives *rewards* from the environment. These are usually a scalar value that can be positive or negative. For example, a learning environment is usually designed such that the goal state has the highest reward and states to be avoided have the lowest.

The agent probabilistically explores the outcome of various actions in various states in order to learn the best way to behave in a given situation (i.e. how to maximize rewards). As it explores the environment the agent maintains a representation of the value of taking a particular action from a given state, this is known as the *Q-value* for that state-action pair. These values are initially random or uniform, and through exploring

the outcome of various actions these Q-values are incrementally updated to more accurately reflect the true value of a particular state-action pair.

1.2.5 The role of the human in standard ML approaches

Standard Machine Learning techniques have had great success in many applications. People have recognized some of the hard problems of learning in the real world, e.g., real-time learning in environments that are partially observable, dynamic, and continuous [Mataric, 1997, Thrun and Mitchell, 1993, Thrun, 2002]. However, learning quickly from interactions with a human teacher poses additional challenges (e.g., limited human patience, ambiguous human input,...). Typically machine learning has not been designed for learning from ordinary human teachers in a real-time social interaction.

Nevertheless, it is always the case that a human is involved in the learning process. The human designer plays an important role in the success of any machine learning system. For example, in their survey of reinforcement learning, [Kaelbling et al., 1996] point out several practical ways that RL algorithms can be biased to improve learning. To illustrate the distinction between an SG-ML approach and the current role of the human in ML, it is useful to look at machine learning from a holistic point of view. What is the role of the machine and what is it that designers have to do to create a successful learning system? In a number of ways the system designer crafts the learning algorithm to learn the right thing at the right time.

- *Data collection*: In the case of pattern recognition systems, collecting the data set with which training and testing takes place is a significant step. The designer must choose a set that is highly representative of the data that the system will see in the future. The size and diversity of the training and testing data set will determine the speed and accuracy of learning and the quality of the resulting system, including its generalization characteristics.
- *Selecting the feature space and its structure*: Deciding what input features and similarity metrics are most important for discriminating in the task and environment at hand is a critical step. For example, in a classification task, the designer must be careful to include input features that are in fact discriminatory and the algorithm will do better if the redundant or non-discriminatory features are excluded. The prior knowledge of the designer about the invariances of the environment plays an important role at this step. Many times input features also need be filtered before being passed to the learning system, designers build these filters to fit the task at hand. This issue of feature choice is not limited to supervised learning techniques. In many of the more successful examples of reinforcement learners,

function approximation techniques are used to learn the value function. In this case, the designer plays a critical role of defining the features that the system will need in order to best calculate its appraisals and represent the environment.

- *Transfer*: Similarly, the underlying representations used in machine learning typically make it difficult for the systems to transfer knowledge learned in one particular setting or task to an alternate setting. The ability to do this type of generalization is highly dependent on representation and feature space decisions made by the designer.
- *Meta-control of the search*: The designer must select the examples and the order in which the system sees the examples, seeding the search for a solution. In many cases, algorithms can suffer from over training, thus another important role the designer plays is that of determining when learning is done.
- *Define a reward signal*: In a reinforcement learning system, a critical role of the designer is defining the reward signal that the agent will receive. In defining this signal the designer defines the task goals for the learning agent (since the RL agent's goal is to maximize reward).
- *Subtasking the problem*: This is specifically a technique used in reinforcement learning, to speed up the learning of a complicated task the designer has the system first learn policies for the subtasks.

Thus, the learning process for standard ML techniques is not currently feasible for non-experts. In Socially Guided Machine Learning, the goal is to understand how to bridge this gap, enabling machine learning systems to succeed at learning within a social interaction with everyday people.

1.3 Related Work

1.3.1 Approaches designed for human input

For years researchers working on robotic and software agents have been inspired by the idea of efficiently transferring knowledge about tasks or skills from a human to a machine. There are several related works that explicitly incorporate human input to a machine learning process. For the large majority of prior work, the evaluation and test scenarios have not used everyday people with these systems. Nonetheless, a review of these works characterizes the ways in which machine learning systems have tried to leverage human input.

Machine learns by observing human behavior

Several prior works have dealt with the scenario where a machine learns by only observing human behavior. In some cases the teaching is implicit, in others the human is explicitly teaching the machine a new skill. In general the SG-ML goal is to have systems that are more interactive than these approaches, that learn in real-time from everyday people and the ways that people will naturally provide demonstrations.

- Personalization agents and adaptive user interfaces rely on the human as an implicit teacher to model human preferences or activities through passive observation of the user's behavior [Lashkari et al., 1994, Horvitz et al., 1998].
- Programming by Example is a technique to allow a person to explicitly teach a software agent [Lieberman, 2001]. For example, the Mondrian system records demonstrated procedures in a graphical user interface and learns a generalized model that can later be used in a similar context.
- In an approach called learning by watching, a robot is able to observe a human demonstrating a blocks assembly task [Kuniyoshi et al., 1994]. From this observation, the system extracts the action sequence and infers a task plan that can be executed by the robot. A very similar approach lets a human wear a glove to demonstrate a peg-in-hole task [Voyles and Khosla, 1998]. The system extracts a high-level state machine of the task that can then be executed on the robot.
- In another approach, a robot uses a human demonstration to learn a reward function for the task [Atkeson and Schaal, 1997]. A human demonstration of the pendulum swing-up task seeds the search for a reward function. Then the system uses Reinforcement Learning to learn a model of the task with the learned reward function.
- A number of works have focused on this notion of skill learning by demonstration or imitation (reviewed in [Schaal, 1999, Breazeal and Scassellati, 2002]). The few examples given here are representative of the work and the nature of human interaction in these approaches. There is generally a specific training phase, where the machine observes the human, then a machine learning technique is used to abstract or infer a model of the demonstrated skill.

Human explicitly directs action of the machine

In other works the human is able to directly influence the actions of the machine to provide it with an experience from which to learn. These approaches are much more in-

teractive than learning by observation approaches and more closely resemble the goals of an SG-ML system. However, for a large majority of these works the human is required to learn how to correctly interact with the machine. Additionally the teacher needs to know precisely how the machine is to perform the task. In some cases the human input portion of the learning interaction amounts to programming the task for the machine.

- In a recent robot task learning example, the robot learns a navigation task by following a human demonstrator [Nicolescu and Matarić, 2003]. The teacher uses simple voice cues to frame the learning (“here,” “take,” “drop,” “stop”), and the robot generalizes a task model over multiple trials with the human.
- Many people have worked on systems for translating natural language communication into a more formal language that can be used to instruct a machine. In a robot learning example, the human teacher uses natural language to instruct a mobile robot in a navigation task [Lauria et al., 2002] (all of the instruction happens prior to execution). Natural language communication has also been leveraged in reinforcement learning systems allowing human teachers to provide domain specific advice to the action selection mechanism [Kuhlmann et al., 2004, Maclin et al., 2005].
- Several works use the notion of supervising a learning agent by directly controlling the training action sequence. Lin developed a way to specify teaching sequences or experience for an RL agent, with the recognition that a human teacher can help the agent efficiently explore the most interesting parts of the state space [Lin, 1992]. Others have achieved similar improvements by letting a human directly control the actions of a robot agent with teleoperation to supervise a RL process [Smart and Kaelbling, 2002], or to provide example task demonstrations [Peters and Campbell, 2003].
- Loosening the burden on the human teacher, other approaches let the human supervise an RL agent by occasionally biasing action selection rather than directly controlling all of the agent’s actions [Clouse and Utgoff, 1992, Kuhlmann et al., 2004, Maclin et al., 2005].

Human provides high-level evaluation, feedback, or labels to a machine learner

In other cases the human influences the experience of the machine with higher level constructs than individual actions, for example, providing feedback to a reinforcement learner or labels to an active learning system.

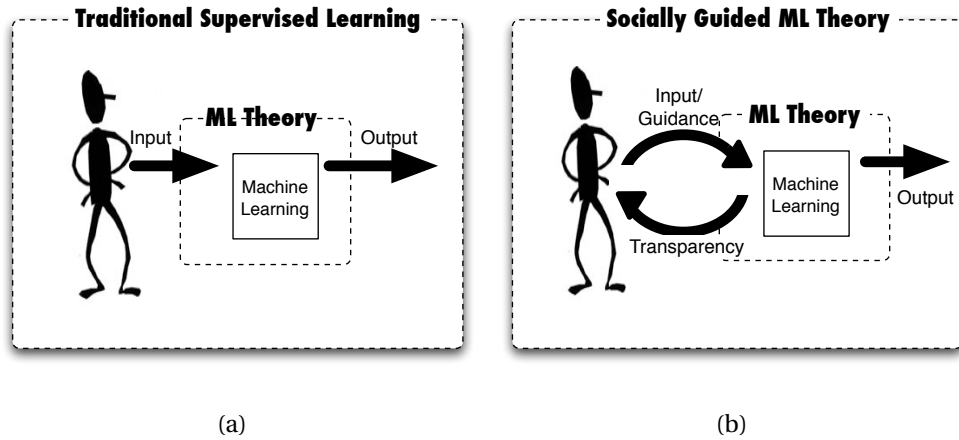


Figure 1-1: SG-ML explicitly acknowledges the human in the loop, in contrast to standard supervised ML techniques.

- Several approaches are inspired by animal training techniques like clicker training and shaping [Blumberg et al., 2002, Kaplan et al., 2002, Saksida et al., 1998]. The main principle behind these approaches is that learning involves reinforcing the connections of base behaviors to a resultant complex behavior, or reinforcing a perceptual-motor association. A human trainer uses instrumental conditioning techniques and signals the agent when a goal behavior has been achieved. Related to this, a common approach for incorporating human input to a reinforcement learner lets the human directly control the reward signal to the agent [Isbell et al., 2001, Evans, 2002, Stern et al., 1998]. In these cases the human can provide positive and negative feedback at any point, rather than only positive feedback according to an instrumental conditioning reward schedule.
- Active learning or learning with queries is an approach that explicitly acknowledges a human in the loop [Cohn et al., 1995, Schohn and Cohn, 2000]. This is a semi-supervised learning approach that utilizes a human 'oracle' through queries. An unsupervised learning algorithm identifies the most interesting examples, and then asks the oracle for labels. Thus, the algorithm is in control of the interaction without regard of what an ordinary human will be able to provide in a real scenario.

1.3.2 An Interaction perspective of ML

In many of the related works mentioned above, the primary motivation for leveraging human input is to achieve some learning performance gains for the machine. In Socially

Guided Machine Learning, we advocate designing for the performance of the complete, coupled human-machine teaching-learning system. This new perspective reframes the machine learning problem as an interaction between the human and the machine. This allows us to take advantage of human teaching behavior to construct a machine learning process that is more amenable to the human partner.

Figure 1-1(a) is a high level view of a supervised machine learning process. A human provides input examples to the learning mechanism, which performs its task and provides some output. Alternatively, an SG-ML view of learning models the complete human-machine system, characterized in Figure 1-1(b).

This simple diagram highlights the key aspects of a social learning system, an interaction approach to machine learning forces the research community to consider many new questions. We need a principled theory of the content and dynamics of this tightly coupled teaching-learning process in order to design systems that can learn efficiently and effectively from ordinary users.

Input Channels: An SG-ML approach begins with the question: “How do humans want to teach?” In addition to designing the interaction based on what the machine needs to succeed in learning, we need to also understand what kinds of intentions people will try to communicate in their everyday teaching behavior. We can then change the input portion of the machine learning training process to better accommodate a human partner.

Output Channels: An SG-ML approach asks: “How can the output provided by the learning agent improve the performance of the teaching-learning system?” In a tightly coupled interaction, a ‘black box’ learning process does nothing to improve the quality and relevance of the instructional guidance. However, transparency of the internal state of the machine could greatly improve the learning experience. By communicating its internal state, revealing what is known and what is unclear, the robot can guide the teaching process. To be most effective, the robot should reveal its internal state in a manner that is intuitive for the human partner [Breazeal, 2002, Arkin et al., 2003].

Input/Output Dynamics: Combining the previous two topics, this topic recognizes that these input and output channels interact over time. The dynamics of the interaction can change the nature of the input from the human. In particular, the temporal structure of teaching versus performing may significantly influence the behavior of the human. An incremental, on-line learning system creates a very different experience for the human than a system that must receive a full set of training examples before its performance can be evaluated. Iterative feedback allows for on-line refinement; the human can provide another example or correct mistakes right away instead of waiting to evaluate the results at the end of the training process. This may provide a significant benefit

to the human's level of engagement and motivation. The sense that progress is being made may keep the human engaged with the training process for a longer period of time, which in turn benefits the learning system.

1.4 Thesis Overview

Socially Guided Machine Learning proposes an alternate view of the machine learning problem, viewing the teaching-learning problem as a collaboration between the machine and the human partner, and using human social skills to constrain and guide the learning process. More than a good interface technique, the ability to utilize and leverage human social structure can positively impact the underlying learning mechanism.

Chapter 2 presents an investigation with a computer game, *Sophie's Kitchen*. An experiment with human subjects provides several insights about how people approach the task of teaching a machine. In particular, people want to direct and guide an agent's exploration process, they quickly use the behavior of the agent to infer a mental model of the learning process, and they utilize positive and negative feedback in asymmetric ways. Chapters 3, 4, and 5 provide an exploration of each of these themes on a robotic platform, Leonardo, and with follow-up studies in the *Sophie's Kitchen* platform. These implementations and experiments show several explicit ways that social interaction can significantly improve the speed, efficiency, and understandability of a machine learning process, making it more successful in a real-time interaction with everyday human trainers:¹

- An experiment investigates human teaching behavior and yields three general characteristics exhibited across participants – Chapter 2.
- The guidance-exploration spectrum is a novel characterization of human interaction with machine learning. Three implementations represent several points along this spectrum – Chapter 3.
- An implementation and experiment in *Sophie's Kitchen* shows that everyday human trainers are able to use guidance with a Reinforcement Learning agent, resulting in significant performance improvements – Chapter 3.
- Novel approaches and implementations of goal-oriented task learning are demonstrated on the Leonardo robot – Chapter 3.

¹Aspects of these thesis contributions have been published in several conference and journal publications: [Thomaz and Breazeal, 2006a, Thomaz and Breazeal, 2006b, Lockerd and Breazeal, 2004, Thomaz et al., 2005b, Breazeal et al., 2004b, Breazeal et al., 2004a, Thomaz et al., 2006, Breazeal et al., 2005b, Thomaz et al., 2005a, Breazeal et al., 2005c]

- Implementations of transparency devices to reveal aspects of the internal learning state are shown with software and robotic agents. Experiments with both Sophie and Leonardo show that transparency leads to significant improvements in the quality of instruction received from a human teacher – Chapter 4.
- Implementations with Sophie and Leonardo represent two asymmetric interpretations of feedback from a human teacher. An experiment with human trainers shows significant positive benefits to the learning mechanism – Chapter 5.

In aiming to enable robots and machines in general to learn new tasks from natural human instruction with ordinary people (not experts in robotics or machine learning), it will be important to enable these systems to take advantage of social interactions. Structuring guidance through interpersonal interaction will be natural for everyday people who need to teach their machines new things — this thesis provides several contributions towards the understanding of Socially Guided Machine Learning, explicating the fundamental SG-ML principles of Guidance, Transparency, Asymmetry, and Goal-Oriented Learning.

Chapter 2

Experiments in Socially Guided Machine Learning

As reviewed in the previous chapter, several examples exist of machines learning from human input, but the role of a human teacher is not adequately understood or leveraged by machine learning systems that are meant to learn from humans. Many of the examples of agents that learn interactively with a human teacher are Reinforcement Learning (RL) based approaches. Reinforcement learning has certain desirable qualities for an SG-ML agent, in particular the general strategy of exploring and learning from experience, and evaluating the world through a reward function. The reward function defines states in the world that are positive, negative, or neutral and is pre-specified by the designer of the algorithm. This enables the agent to learn in an unsupervised fashion through its own experience.

Although the theory of reinforcement learning was originally formulated for systems to learn on-line, independent of human participation, the algorithm is amenable to incorporating real-time human feedback by having a person supply reward and/or punishment as an additional input to the reward function. This has been a popular technique for letting humans teach robots and game characters new skills [Blumberg et al., 2002, Kaplan et al., 2002, Isbell et al., 2001, Evans, 2002, Stern et al., 1998]. This assumption models the human input as indistinguishable from any other feedback coming from the environment, and assumes that people's communication will concern only feedback on past actions. *But are these good assumptions?*

Reinforcement-based learning approaches need to be reformulated to more effectively incorporate a human teacher (that is not an expert in machine learning). To do this properly, we must deeply understand the human teacher's contribution: *how* does the human teach, and *what* are they trying to communicate to the learner? For instance, how do people actually use a reward signal? Do they only use it as a feedback signal to

reinforce the last action the agent performed, or do they also use it to guide the agent's next action as a sort of anticipatory reward? Furthermore, if the reward channel has a dual use in practice, then does the agent's learning algorithm properly distinguish this information to take advantage of it? In general, the human's role in teaching as real-time interaction has been a neglected topic.

This chapter presents a systematic study and analysis of human behavior when teaching a virtual graphical character to perform a novel task within a reinforcement-based learning framework. The experimental system, *Sophie's Kitchen*, is a computer game that allows an agent to be trained interactively to bake a cake through sending the agent feedback messages. An experiment with human subjects finds several prominent characteristics for how human players approach the task of explicitly teaching a learning agent.

- People want the ability to direct the agent's attention, guiding the exploration.
- Players try to maximize their impact on the learning process as they infer a model of the learner, suggesting that transparency behaviors that reveal the internal state of the agent, such as gaze, can be utilized to improve the human's teaching.
- Positive and negative feedback from a human teacher have asymmetrical intentions or meanings.

2.1 The *Sophie's Kitchen* Platform

Sophie's Kitchen is a Java-based computer game platform, designed to investigate how human interaction can and should change the machine learning process. *Sophie's Kitchen* is an object-based state-action MDP space for a single agent using a fixed set of actions on a fixed set of objects.

2.1.1 Sophie's MDP

The task scenario used is a kitchen world (see Fig. 2-1), where the agent (Sophie) learns to bake a cake. This system is defined by (L, O, Σ, T, A) .

- There are a finite set of k locations $L = \{l_1, \dots, l_k\}$. In the kitchen task scenario $k = 4$; $L = \{\text{Shelf}, \text{Table}, \text{Oven}, \text{Agent}\}$. As shown in Fig. 2-1, the agent is in the center surrounded by a shelf, table and oven; and the location *Agent* is available to objects (i.e., when the agent picks up an object, then it has location *Agent*).



Figure 2-1: *Sophie's Kitchen*. The agent is in the center, with a shelf on the right, oven on the left, a table in between, and five cake baking objects. The vertical bar is the interactive reward and is controlled by the human.

- There is a finite set of n objects $O = \{o_1, \dots, o_n\}$. Each object can be in one of an object-specific number of mutually exclusive object states. Thus, Ω_i is the set of states for object o_i , and $O^* = (\Omega_1 \times \dots \times \Omega_n)$ is the entire object configuration space. In the kitchen task scenario $n = 5$: the objects Flour, Eggs, and Spoon each have only one object state; the object Bowl has five object states: empty, flour, eggs, both, mixed; and the object Tray has three object states: empty, batter, baked.
- Let L^A be the possible agent locations: $L^A = \{\text{Shelf}, \text{Table}, \text{Oven}\}$; and let L^O be the possible object locations: $L^O = \{\text{Shelf}, \text{Table}, \text{Oven}, \text{Agent}\}$. Then the legal set of states is $\Sigma \subset (L^A \times L^O \times O^*)$, and a specific state is defined by $(l_a, l_{o_1} \dots l_{o_n}, \omega)$: the agent's location, $l_a \in L^A$, and each object's location, $l_{o_i} \in L^O$, and the object configuration, $\omega \in O^*$.
- T is a transition function: $\Sigma \times A \mapsto \Sigma$. The action space A is expanded from four atomic actions ($\text{GO}\langle x \rangle$, $\text{PUT-DOWN}\langle x \rangle$, $\text{PICK-UP}\langle x \rangle$, $\text{USE}\langle x \rangle\langle y \rangle$): Assuming the locations L^A are arranged in a ring, the agent can always GO left or right to change location; she can PICK-UP any object in her current location; she can PUT-DOWN any object in her possession; and she can USE any object in her possession on any object in her current location. The agent can hold only one object at a time. Thus the set of actions available at a particular time is dependent on the particular state, and is a subset of the entire action space, A . Executing an action advances the world state in a deterministic way defined by T . For example, executing $\text{PICK-UP}\langle \text{Flour} \rangle$ advances the state of the world such that the Flour has location Agent. USEing an ingredient on the Bowl puts that ingredient in it; using the Spoon on the

Algorithm 1 Q-Learning with Interactive Rewards from a Human Partner

```
1:  $s$  =last state,  $s'$  =current state,  $a$  =last action,  $r$  =reward  
2: while learning do  
3:    $a$  = random select weighted by  $Q[s, a]$  values  
4:   execute  $a$ , and transition to  $s'$   
      (small delay to allow for human reward)  
5:   sense reward,  $r$   
6:   update Q-value:
```

$$Q[s, a] \leftarrow Q[s, a] + \alpha(r + \gamma(\max_{a'} Q[s', a']) - Q[s, a])$$

```
7: end while
```

both-Bowl transitions its state to the mixed-Bowl, etc.

In the initial state, s_0 , all objects and the agent are at location Shelf. A successful completion of the task will include putting flour and eggs in the bowl, stirring the ingredients using the spoon, then transferring the batter into the tray, and finally putting the tray in the oven. Some end states are so-called *disaster* states (for example—putting the eggs in the oven), which result in a negative reward ($r = -1$), the termination of the current trial, and a transition to state s_0 . In order to encourage short sequences, an inherent negative reward of $r = -.04$ is placed in any non-goal state.

Due to the flexibility of the task, there are many action sequences that can lead to the desired goal. Here is one such sequence:

PICK-UP Bowl; GO right; PUT-DOWN Bowl; GO left; PICK-UP Flour; GO right; USE Flour, Bowl; PUT-DOWN Flour; GO left; PICK-UP Eggs; GO right; USE Eggs, Bowl; PUT-DOWN Eggs; GO left; PICK-UP Spoon; GO right; USE Spoon, Bowl; PUT-DOWN Spoon; GO left; PICK-UP Tray; GO right; PUT-DOWN Tray; PICK-UP Bowl; USE Bowl, Tray; PUT-DOWN Bowl; PICK-UP Tray; GO right; PUT-DOWN Tray.

2.1.2 Learning Algorithm

The algorithm implemented for the experiments presented in this chapter is a standard Q-Learning algorithm (learning rate $\alpha = .3$ and discount factor $\gamma = .75$) [Watkins and Dayan, 1992]. This is shown above in Algorithm 1. A slight delay happens in line 4 as the agent's action is animated and also to allow the human time to issue interactive rewards. Q-Learning is used as the instrument for this work because it is a widely understood RL algorithm, thus affording the transfer of these lessons to other reinforcement-based approaches.

2.1.3 Interactive Rewards Interface

A central feature of *Sophie's Kitchen* is the interactive reward interface. Using the mouse, a human trainer can—at any point in the operation of the agent—award a scalar reward signal $r \in [-1, 1]$. The user receives visual feedback enabling them to tune the reward signal before sending it to the agent. Choosing and sending the reward does not halt the progress of the agent, which runs asynchronously to the interactive human reward.

The interface also lets the user make a distinction between rewarding the whole state of the world or the state of a particular object (object specific rewards). An object specific reward is administered by doing a feedback message on a particular object (objects are highlighted when the mouse is over them to indicate that any subsequent reward will be object specific). This distinction exists to test a hypothesis that people will prefer to communicate feedback about particular aspects of a state rather than the entire state. However, object specific rewards are used only to learn about the human trainer's behavior and communicative intent; the learning algorithm treats all rewards in the traditional sense of pertaining to a whole state and action pair.

2.2 Experimental Design

The purpose of this initial experiment with *Sophie's Kitchen* is to understand, when given a single reward channel (as in prior works), how do people use it to teach the agent? In the experiment, 18 participants played a computer game, in which their goal was to get the virtual robot, Sophie, to learn how to bake a cake on her own. Participants were asked to rate their expertise with machine learning software and systems on a scale of 1 to 7, (1=no experience, 7=very experienced), and we found it was an above average but reasonably diverse population (mean=3.7; standard deviation=2.3).¹

Participants were told they could not tell Sophie what to do, nor could they do actions directly, but they could send Sophie the following messages via a mouse to help her learn the task:

- Click and drag the mouse up to make a green box, a positive message; and down for red/negative (Figure 2-1 shows a positive feedback message).
- By lifting the mouse button, the message is sent to Sophie, she sees the color and size of the message.
- Clicking on an object, this tells Sophie your message is about that object. As in, “Hey Sophie, this is what I’m talking about...” If you click anywhere else, Sophie assumes your feedback pertains to everything in general.

¹We had both male and female participants, but did not keep gender statistics of the population.

The system maintains an activity log and records time step and real time of each of the following: state transitions, actions, human rewards, reward aboutness (if object specific), disasters, and goals. Additionally, there was an informal interview after subjects completed the task.²

2.3 Findings

2.3.1 Guidance Intentions

Even though the instructions clearly stated that communication of both general and object specific rewards were *feedback* messages, many people assumed that object specific rewards were future directed messages or guidance for the agent. Several people mentioned this in the interview, and this is also suggested through behavioral evidence in the game logs.

An object reward used in a standard RL sense, should pertain to the last object the agent used. Figure 2-2 has a mark for each player, indicating the percentage of object specific rewards that were about the last object the agent used: 100% would indicate that the player always used object rewards in a feedback connotation, and 0% would mean they never used object rewards as feedback. We can see that several players had object rewards that were rarely correlated to the last object (i.e., for 8 people less than 50% of their object rewards were about the last object).

Interview responses suggested these people's rewards actually pertain to the future, indicating what they want (or do not want) the agent to use next. A single test case is used to show how many people used object rewards as a guidance mechanism: When the agent is facing the shelf, a guidance reward could be administered (i.e., what to pick up). Further, a positive reward given to either the empty bowl or empty tray on the shelf could *only* be interpreted as guidance since this state would not be part of any desired sequence of the task (only the initial state). Thus, rewards to empty bowls and trays in this configuration serve to measure the prevalence of guidance behavior.

Figure 2-3 indicates how many people tried giving rewards to the bowl or tray when they were empty on the shelf. Nearly all of the participants, 15 of 18, gave rewards to the bowl or tray objects sitting empty on the shelf. This leads to the conclusion that many participants tried using the reward channel to guide the agent's behavior to particular objects, giving rewards for actions the agent was *about to do* in addition to the traditional rewards for what the agent had just done.

These *anticipatory* rewards observed from everyday human trainers will require new

²The full protocol, instructions and consent form used in the study can be found in Appendix A.

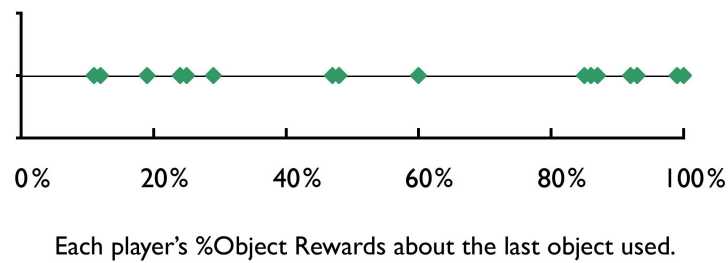


Figure 2-2: There is one mark for each player, indicating their percentage of object rewards that were about the last object of attention. This graph shows that many people had object rewards that were rarely about the last object, thus rarely used in a feedback orientation.



Figure 2-3: A reward to the empty bowl or tray on the shelf is assumed to be meant as guidance instead of feedback. This graph shows that 15 of the 18 players gave rewards to the bowl/tray empty on the shelf.

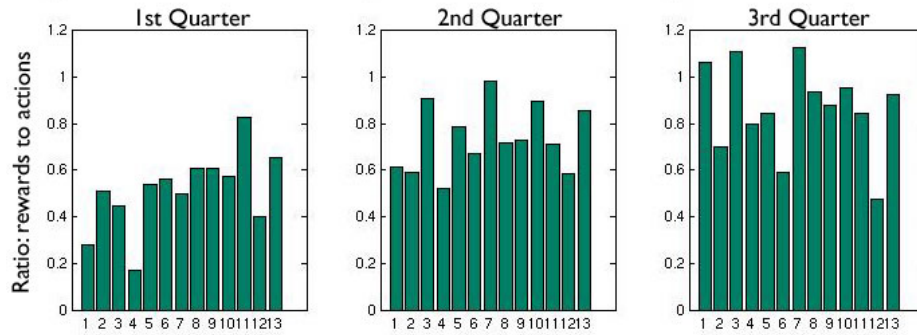


Figure 2-4: Ratio of rewards to actions over the first three quarters of the training sessions shows an increasing trend.

attention in learning systems and algorithms in order for agents to correctly interpret their human partners. Chapter 3 covers the design, implementation, and evaluation of various techniques for utilizing social guidance in a machine learning system.

2.3.2 Inferring a Model of the Learner

In human learning, teachers direct a learner's attention, structure experiences, support attempts, and regulate complexity. The learner contributes by revealing their internal state to help guide the teaching process. Each simplifies the task for each other. This *collaborative* aspect of teaching and learning has been stressed in prior work [Breazeal et al., 2004a], and the findings in this study support this notion of *partnership*. When everyday users are asked to train a machine learning agent, they adjust their training behavior as the interaction proceeds, reacting to the behavior of the learner.

Informed by related work [Isbell et al., 2001], it is reasonable to expect people would habituate to the activity and that feedback would decrease over the training session. However, just the opposite was found: the ratio of rewards to actions over the entire training session had a mean of .77 and standard deviation of .18. Additionally, there is an increasing trend in the rewards-to-actions ratio over the first three quarters of training. Fig. 2-4 shows data for the first three quarters for training, each graph has one bar for each individual indicating the ratio of rewards to actions. A 1:1 ratio in this case means that the human teacher gives a reward after every action taken by the agent. By the third graph more bars are approaching or surpassing a ratio of 1.

One explanation for this increasing trend is a shift in mental model; as people realize the impact of their feedback they adjusted their reward schedule to fit this model of the learner. This finds anecdotal support in the interview responses. Many users reported that at some point they came to the conclusion that their feedback was helping the agent

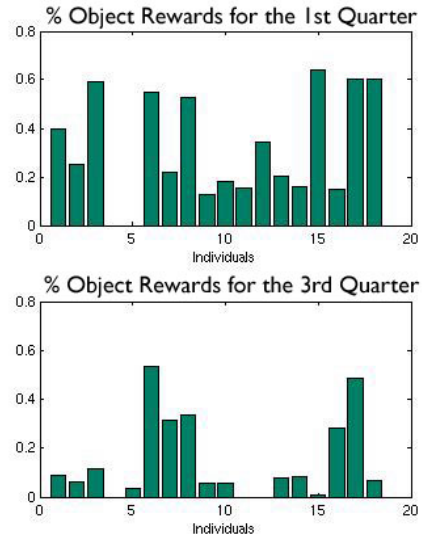


Figure 2-5: Each bar represents an individual and the height is the percentage of object rewards. The difference in the first and last training quarters shows a drop off in usage over time.

learn and they subsequently gave more rewards. Many users described the agent as a “stage” learner, that it would seem to make large improvements all at once. This is precisely the behavior one sees with a Q-Learning agent: fairly random exploration initially, and the results of learning are not seen until the agent restarts after a failure. Without any particular understanding of the algorithm, participants were quickly able to develop a reasonable mental model of the agent through the interaction. They were encouraged by the learning progress, and subsequently gave more rewards.

A second expectation was that people would naturally use goal-oriented and intentional communication (measured by allowing people to specify object specific rewards, explained in Sec. 2.1.3). The difference between the first and last quarters of training shows that many people tried the object specific rewards at first but stopped using them over time. In the interview, many users reported that the object rewards “did not seem to be working.” Thus, many participants tried the object specific rewards initially, but were able to detect over time that an object specific reward did not have a different effect on the learning process than a general reward (which is true), and therefore stopped using the object rewards.

These are concrete examples of the human trainer’s propensity to learn from the agent how to best impact the process. This presents a huge opportunity for an interactive learning agent to *improve its own learning environment* by communicating more internal state to the human teacher, making the learning process more transparent. Chapter 4 details the use of transparent behavior to improve a learning environment.

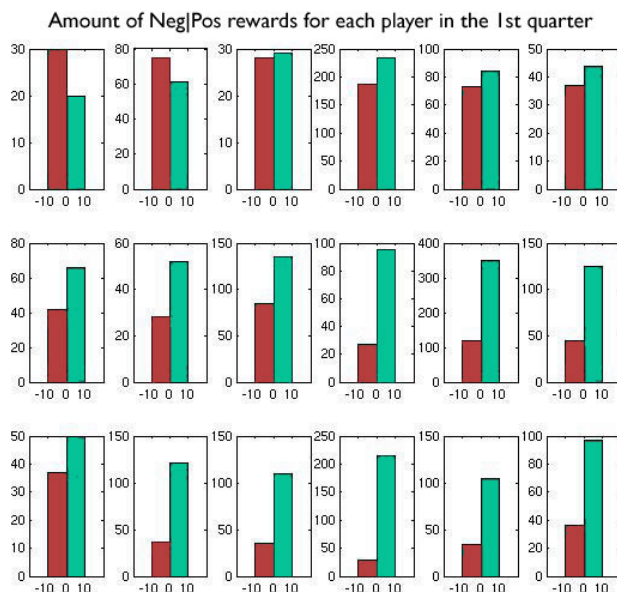


Figure 2-6: Histograms of rewards for each individual in the first quarter of their session. The left column is negative rewards and the right is positive rewards. Most people even in the first quarter of training have a much higher bar on the right.

2.3.3 An Asymmetric Use of Rewards

For many people, a large majority of rewards given were positive, the mean percentage of positive rewards for players was 69.8%. This was thought at first to be due to the agent improving and exhibiting more correct behavior over time (soliciting more positive rewards); however, the data from the first quarter of training shows that well before the agent is behaving correctly, the majority of participants still show a positive bias. Fig. 2-6 shows reward histograms for each participant's first quarter of training; the number of negative rewards on the left and positive rewards on the right, most participants have a much larger bar on the right. A plausible hypothesis is that people are falling into a natural teaching interaction with the agent, treating it as a social entity that needs encouragement. Some people specifically mentioned in the interview that they felt positive feedback would be better for learning. Chapter 5 is devoted to the investigation of asymmetric interpretations of human feedback for machine learning systems.

Chapter 3

Utilizing Social Guidance

The aim of SG-ML is to have a system that learns new tasks in partnership with a human, in a way that is intuitive for the human teacher. The *Sophie's Kitchen* experiment in the last chapter showed people's desire to guide and direct the agent in the learning task. This chapter investigates various forms of social guidance for machine learning systems.

An important research theme that this chapter addresses is the spectrum of guidance and exploration. As seen in prior works (Sec. 1.3.1) most systems that incorporate a human teacher into the learning process maintain a constant level of involvement of the human partner. Several are highly dependent on the human teacher's guidance, and will learn nothing without their interaction. Others are almost entirely exploration based, and barely take advantage of the human partner. An important research question for SG-ML is how to seamlessly incorporate both guidance and exploration, resulting in a system that can learn on its own, but also take full advantage of a human partner if they are there to provide guidance.

The systems in this chapter represent three points along this spectrum. The first learning system presented is 'Learning within a Social Dialog' on the Leonardo robot. This implementation has many desirable SG-ML qualities that allow it to take advantage of natural human guidance within a tutorial dialog. This guidance-heavy system is followed with the presentation of a highly exploration based learner: the *Sophie's Kitchen* game modified to incorporate human guidance. A second experiment with human subjects allows us to quantify the effects of guidance on a standard exploratory learner. Finally, the lessons from these two systems are incorporated into a third learning mechanism, 'Guided Exploration', implemented on the Leonardo robot.

A second important theme of this chapter is goal-oriented learning. In many prior works in which a machine learns a new task or skill, there is an assumption that the goal is somehow defined by the designer, or the goal is to learn a complete world model. Alternatively, both the Social Dialog and the Guided Exploration implementations do

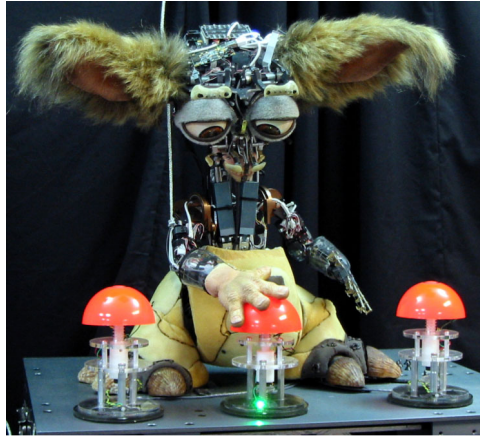


Figure 3-1: Leo and his workspace with three button toys.

not make this assumption. Instead, these two approaches let the systems learn new tasks/goals with a human partner. A goal-oriented approach to learning is a fundamental capability necessary for social learners. Given that their social partners will act and interpret action in intentional and goal-oriented ways, an SG-ML system will need to continually work to refine the concept of what the human partner is meaning to communicate, and what the activity is *about*.

3.1 The Leonardo Robot Platform

The second research platform used in this thesis, in addition to *Sophie's Kitchen*, is Leonardo ("Leo"), a humanoid robot with 65 degrees of freedom that has been specifically designed for social interaction using a range of facial and body pose expressions (see Figure 3-1). Leonardo has been under development in the Robotic Life Group of the MIT Media Lab since 2002, and is a collaboration with Stan Winston Studios. This section briefly introduces aspects of the Leonardo architecture necessary to understand the social learning capabilities. For more specific details on the robotic platform refer to the following: [Breazeal et al., 2004a, Breazeal et al., 2005a, Gray et al., 2005, Hancher, 2003].

3.1.1 Sensory Inputs

Leo has both speech and vision sensory inputs and relies on gestures and facial expression for social communication. Leo sees the world through two environmentally mounted stereo-vision cameras. One stereo camera is mounted behind Leo's head for detecting humans within the robot's interpersonal space (within approximately 4 feet of the robot) and determining their head pose [Morency et al., 2002]. The second stereo

camera looks down from above, and detects objects in Leo's space as well as human hands pointing to these objects [Brooks and Breazeal, 2006]. Leo can use his eye cameras for fine corrections to look directly at objects or faces and to view them at a higher resolution.

The speech understanding system is based on the Sphinx system [Lamere et al., 2003]. The system has a limited grammar to facilitate accuracy of the voice recognition, and it parses recognized phrases into symbols that are sent to the cognitive system.¹

3.1.2 Cognitive Architecture

The cognitive system extends the C5M architecture, a recent version of the C4 system described in [Blumberg et al., 2001]. As a foundation of the learning implementations presented in this chapter, this section presents a technical description of two components of Leo's cognitive architecture: the Perception System and the Belief System.² The Perception System is responsible for extracting perceptual features from raw sensory information, and the Belief System is responsible for integrating this information into discrete object representations. The Belief System represents our approach to sensor fusion, object tracking and persistence.

On every time step, the robot receives a set of sensory observations $O = \{o_1, o_2, \dots, o_M\}$ from its various sensory processes. As an example, imagine that the robot receives information about button toys and their locations from an eye-mounted camera, and information about the state of a light on the buttons from an overhead camera. On a particular time step, the robot might receive the observations $O = \{(\text{red object at } (10,0,0)), (\text{button object at position } (10,0,0)), (\text{green object at } (0,0,0)), (\text{button object at } (0,0,0)), (\text{blue object at } (-10,0,0)), (\text{button object at } (-10,0,0)), (\text{light at } (10,0,0)), (\text{light at } (-10,0,0))\}$.

Information is extracted from these observations by the Perception System, which consists of a set of *percepts* $P = \{p_1, p_2, \dots, p_K\}$. Each $p \in P$ is a classification function defined such that

$$p(o) = (m, c, d)$$

where $m, c \in [0, 1]$ are match and confidence values and d is an optional derived feature value. For each observation $o_i \in O$, the Perception System produces a *percept snapshot*

$$n_i = \{(p, m, c, d) | p \in P, p(o_i) = (m, c, d), m * c > k\}$$

where $k \in [0, 1]$ is a threshold value, typically 0.5. Returning to our example, the robot might have four percepts relevant to the buttons and their states: a location percept, a

¹The full grammar used with Leonardo can be found in Appendix B

²These technical details are reiterated from [Breazeal et al., 2005b] for the reader's convenience.

color percept, a button shape recognition percept, and a button light recognition percept. The Perception System would produce eight percept snapshots corresponding to the eight sensory observations, containing entries for relevant matching percepts.

These snapshots are then clustered into discrete object representations called *beliefs* by the Belief System. This clustering is typically based on the spatial relationships between the various observations, in conjunction with other metrics of similarity. The Belief System maintains a set of beliefs B , where each belief $b \in B$ is a set mapping percepts to history functions: $b = \{(p_1, y_1), (p_2, y_2), \dots\}$. For each $(p, y) \in b$, y is a history function defined such that

$$y(t) = (m'_t, c'_t, d'_t)$$

represents the “remembered” evaluation for percept p at time t .

The Belief System manages three key processes: creating new beliefs from incoming percept snapshots, merging sets of beliefs, and culling stale beliefs. For the merging process, the Belief System has a number of relevant distance metrics, including a measure of Euclidean spatial distance along with a number of metrics based on symbolic feature similarity (e.g., a symbolic metric might judge observations that are hand-shaped as distant from observations that are button-shaped, thus separating these observations into distinct beliefs even if they are collocated). Returning again to our example, the merge process would produce three beliefs from the original eight sensory observations (merging by spatial location in this case): a red button in the ON state, a green button in the OFF state, and a blue button in the ON state.

The work in this thesis builds on these existing processing modules, adding higher-level cognitive capabilities for representing and learning goal-oriented tasks, motivated exploratory behavior, and expression and gesture capabilities to support a natural collaborative dialog with a human teacher.

3.2 A Socially Guided Learning Dialog

The first guided learning mechanism is an implementation on the Leonardo platform for social learning within a collaborative dialog with a human teacher. Task and goal representations are initially learned with the help of the human and continue to be refined in subsequent executions of the task. In the learning scenario the human stands opposite of Leo in his workspace (pictured in Figure 3-1), and they use speech and gestures to help Leonardo build representations of new tasks/skills based on an initial set of primitive known actions (pointing, pressing, looking).

The *Task Learning Module* maintains the collection of known tasks and arbitrates between task learning and execution, the functionality of this module is illustrated in Fig-

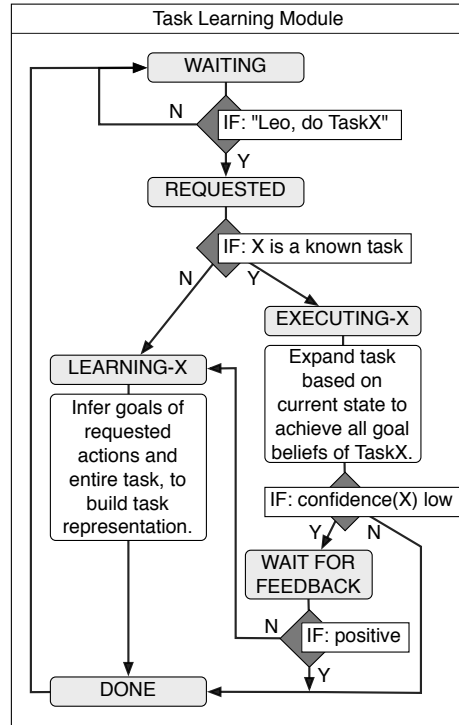


Figure 3-2: An overview of the states and flow of execution in the Task Learning Module, which allows Leo to learn from a human partner within a social dialog.

Figure 3-2. It continually listens for a task-related request from the human partner. Upon encountering a task-related request from the human partner (e.g., “Leo, do *task x*”, “Leo, can you do *task x*?”, etc.) the Task Learning Module enters either the learning or the execution state, and answers the person (using head nods and shakes) if the request was a question.

The Task Learning Module maintains a collection of known tasks. If Leo is asked to do a task that he already knows, then the Task Learning Module executes it by expanding the task’s action and sub-tasks onto a *focus stack* (in a similar way to [Grosz and Sidner, 1990]). The Task Learning Module proceeds through the actions on the stack popping them as they are done or, for a sub-task, pushing its constituent actions onto the stack.

Alternatively, when an unknown task is requested, Leo starts the learning process by indicating that he does not know, shrugging his shoulders and making a confused facial expression. The human partner can then offer to teach the task (“I can teach you to X...”). At this point Leo will confirm with a head nod and the learning process has begun. This exchange is particularly important since it initiates the learning process and establishes a mutual belief about the roles of teacher and learner.

Once learning begins, the human walks the robot through the components of the task, requesting it to perform the necessary steps to reach the goal, building a new task

from its set of known actions and tasks. While in learning mode, the Task Learning Module continually pays attention to what actions the robot is being asked to perform, encoding the inferred goals with these actions. In order to encode the goal state of a performed action or task, Leo compares the world state before and after its execution. In the case that this action or task caused a change of state, this change is taken to be the state-change goal. Otherwise, the goal is assumed to be of the just-do-it type (i.e., the goal is to perform the actions rather than achieve a particular world state). This produces a hierarchical task representation, where a goal is encoded for each individual part of the task as well as for the overall task. When the human indicates that the task is done, it is added to the Task Learning Module’s collection of known tasks.

Learning is handled recursively, such that a sub-task can be learned within a larger task. If the Task Learning Module receives an additional unknown sub-task request, while learning a task, the current learning process is pushed onto a stack and an additional learning thread is started. Once the sub-task learning is complete, it is popped from the stack and its resulting task is added both to the previous learning process and to the Task Learning Module’s list of known tasks. The original learning process continues, with the newly learned sub-task as part of its task representation.

The following sections give technical details of how tasks and goals are represented, the learning mechanism, the generalization mechanism, and the execution mechanism.

3.2.1 Task Representation

Humans are biased to use an intention-based psychology to interpret another agent’s actions [Dennett, 1987]. Moreover, it has been shown repeatedly that, even from a very young age, we interpret intentions and actions based on goals rather than specific activities or motion trajectories [Woodward et al., 2001, Gleissner et al., 2000, Baldwin and Baird, 2001]. A goal-centric view is particularly crucial in a collaborative task setting, in which goals provide a common ground for communication and interaction. All of this suggests that goals and a commitment to their successful completion should be central to task representation.

Goal Types

To support this idea, we have extended the notion of the C5M *action-tuple* data structure. An action-tuple is a set of preconditions, executables, and until-conditions [Blumberg et al., 2001]. Tasks and their constituent actions are variations of this action-tuple structure with the added notion of *goals*.

As the robot learns a new task, it must learn the goals associated with each action,

each sub-task, and the overall task. The system currently distinguishes between two types of goals: (a) state-change goals that represent a change in the world, and (b) just-do-it goals that need to be executed regardless of their impact on the world. These two types of goals differ in both their evaluation as preconditions and in their evaluation as until-conditions. As part of a precondition, a state-change goal must be evaluated before doing the activity to determine if it is needed. As an until-condition, the robot shows commitment towards the state-change goal in trying to execute the action, over multiple attempts if necessary, until succeeding to bring about the desired state. This commitment to the successful completion of goals is an important aspect of intentional behavior [Bratman, 1992, Cohen and Levesque, 1991]. A just-do-it goal on the other hand will lead to an action regardless of the world state, and will only be performed once.

Hierarchical Tasks & Goals

Tasks are represented in a hierarchical structure of actions and sub-tasks (recursively defined in the same fashion). Since tasks, sub-tasks, and actions are derived from the same action-tuple data structure, they are easily used in a unified way, naturally affording a tree representation for tasks.

When learning a task, a goal is associated with the overall task in addition to each of the constituent actions. Overall task and sub-task goals are distinct from the mere conjunction of the goals of their actions and sub-tasks, and are learned separately. For example, consider a task with two constituent actions, but where the task goal is *not* merely the sum of the constituent goals of these actions. The first action causes a change in the world (the system therefore associates a state-change goal with it), and the second action reverses that change (therefore also having a state-change goal). The overall task goal, however, does not have a *net* state change and therefore becomes a just-do-it goal even though its constituent actions both have state-change goals.

When executing a task, goals as preconditions and until-conditions of actions or sub-tasks manage the flow of decision making throughout the task execution process. Overall task goals are evaluated separately from their constituent action goals to determine whether they need to be executed, as well as checking for completion of a task.

One advantage of this top-level evaluation approach is that it is more efficient than having to poll each of the constituent action goals explicitly. Moreover, this goal-oriented implementation supports a more realistic groundwork for intentional understanding—i.e., to perform the task in a way that accomplishes the *overall intent*, rather than just mechanically going through the motions of performing the constituent actions.

The following specifies the task and goal representation of the Task Learning Module:

- Let $A = \{a_1, \dots, a_i\}$ be the set of Leo's primitive actions. Many actions can be applied to an object in the world (e.g., `point-at`, `referent object`). In this case, let the object be referred to as the `object of attention`. For example, `press button 1` and `press button 2` have the same primitive action and different objects of attention.
- Let $Tasks = \{T_1, \dots, T_j\}$ be the Task Learning Module's set of known tasks.
- Each $T \in Tasks$ is defined by $(\{h_1, \dots, h_n\}, k)$. A set of hypothesis task representations, $\{h_1, \dots, h_n\}$, and a variable, $k \in [1, n]$, indicating the index of the current primary hypothesis.
- Each $h \in T$ is a hypothesis representation of the task T and is defined by (E, G, f) . These define the executables of the task, E , the overall goal of the task, G , and the number of examples seen for this task, f , that are consistent with this hypothesis.
- The set of executables $E = \{(e_1, G_1), \dots, (e_m, G_m)\}$. Each $e \in E$ is either a primitive action $a \in A$ or a subtask $T \in Tasks$, and G_i is the goal of executable e_i .
- Goals for actions and tasks consist of a set of *goal beliefs* about what must hold true in order to consider this action or task achieved. A goal $G = \{x_1, \dots, x_y\}$ where each $x \in G$ is a goal belief.
- If G is not a just-do-it goal, it contains a goal belief for each object that changed over the action or task. Recall from Section 3.1.2, that the Belief System maintains one belief for each object in the world. Goal beliefs are derived from this set of beliefs about objects in the world. Rather than containing a single set of percept values, a goal belief represents a desired change to an object during an action or task by grouping a belief's percepts into *expectation* percepts (indicating an expected object feature value), and *criteria* percepts (indicating which beliefs are relevant to apply this expectation to). Thus, $\forall x \in G, x = \{crit, expt\}$, where $crit = \{p_1, \dots, p_{ct}\}$ and $expt = \{p_1, \dots, p_{ex}\}$. The sets *expt* and *crit* are mutually exclusive.

3.2.2 Learning Mechanism

This section provides technical detail of how the Task Learning Module first creates a new task $T_{new} \in Tasks$. Let t indicate time; then, $t = 0$ is the time that the human initiates the learning process, and $t = end$ is the time the human indicates the task is finished.

Let s_t be the state of the world at time t (i.e. the state of the Belief System at t , thus s_t is a set of belief objects each of which contains the values every percept had at the particular time t)

From time $t = 0 \dots \text{end}$, the Task Learning Module pays attention to the actions $a \in A$ that the human is requesting the robot to do and infers goals for each action in order to build the initial task hypothesis $h_1 \in T_{\text{new}}$. When a requested action is completed at a particular time $t = j \in [0, \text{end}]$ (let this action be a_j), then let $t = i \in [0, j]$ be the time that the most recent action prior to a_j was completed, or 0 if a_j is the first action of T_{new} .

The Task Learning Module creates an executable (e, G) about action a_j : $e = a_j$, G is the set of goal beliefs that represent the state change from $s_i \rightarrow s_j$. Then (e, G) is added to E of h_1 . The procedure for making a goal state, G , given the two states, s_i and s_j is the following: Create a goal belief, x , for each belief in s_i that changed over $s_i \rightarrow s_j$: $\forall b_i \in s_i$ find the corresponding³ belief, $b_j \in s_j$. If there are any percepts differences between b_i and b_j then make a goal belief x in the following way: $\forall p \in b_i$ if b_j has the same value for p then add p to x as a criteria percept (i.e. add p to $\text{crit} \in x$), otherwise add the b_j value of p to x as an expectation percept (i.e. add p to $\text{expt} \in x$). When complete add x to the the set of goal beliefs, G . At the end of this process, G contains a goal belief for each object that incurred any change over $s_i \rightarrow s_j$.

At time $t = \text{end}$, this same process works to infer the overall goal, G , for T_{new} , making the goal inference from the changes over $s_0 \rightarrow s_{\text{end}}$. Now the initial hypothesis h_1 contains the set of executables, E , and the goal G for T_{new} . The goal inference mechanism notes all specific changes that occurred over the task; however, there may still be ambiguity around which aspects of the state change are the goal (the change to an object, a class of objects, the whole world state, etc.). To deal with this ambiguity the system expands a hypothesis space of task representations that are consistent with the seen task. Then hypothesis testing coupled with human interaction disambiguates the overall task goal over a few examples.

3.2.3 Hypothesis Expansion and Generalization

Continuing with the example of creating a new task, T_{new} , once the human indicates that the current task is done, then T_{new} contains one hypothesis of the seen example ($h_1 = (E, G, f)$, where $f = 1$). The Task Learning Module uses h_1 to expand other hypotheses about the desired goal state to yield a hypotheses space of all goal representations consistent with the current demonstration.

³“Corresponding” here refers to the fact that b_i and b_j are actually snapshots from the same belief objects in the Belief System. Recall that beliefs are collections of percept histories, thus b_i and b_j are different timeslices of the same collections of percept histories.

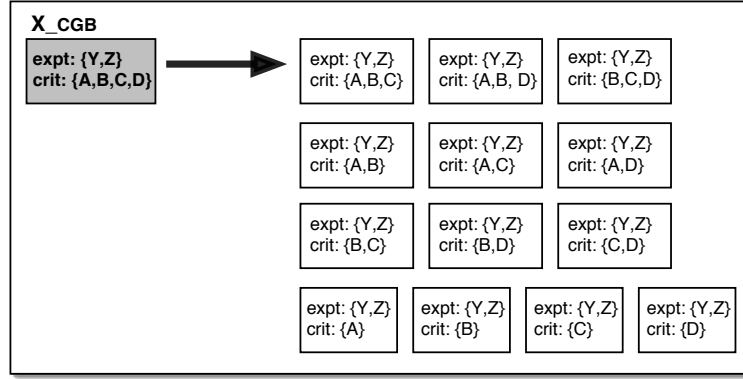


Figure 3-3: The hypothesis space of goal beliefs expanded from the common goal belief x_{CGB} with two expectation features $\{Y, Z\}$, and four criteria features $\{A, B, C, D\}$.

This is similar to a version space of the goal concepts consistent with the demonstration [Buchanan and Mitchell, 1978]. In a version space approach, there is a lattice of hypotheses consistent with the positive examples ordered from most specific to most general. Learning happens through a hypothesis elimination process as more examples of the concept are seen. A primary difference between version spaces learning and the learning presented here is that Leo does not eliminate a hypothesis from the hypothesis space until it is used for execution and fails to achieve the task.

To expand the hypothesis space after a demonstration completes, first the system checks for similarity in the actions performed for this task—i.e. all of the actions $e \in E$ are of the same primitive type $a \in A$ but just have different objects of attention. If this is the case, the primitive action a is noted as the generalized task action. Next the system looks at each of the goal beliefs $x \in G$ of T_{new} (each of the objects that incurred some change) and collapses these into a single common goal belief, x_{CGB} , containing the features common to all. Thus, $x_{CGB} = \{crit_{CGB}, expt_{CGB}\}$ such that each $p \in crit_{CGB}$ is contained in the $crit$ of every $x \in G$ and each $p \in expt_{CGB}$ is contained in the $expt$ of every $x \in G$.

If the sets $crit_{CGB}$ and $expt_{CGB}$ are not empty, then a number of task hypotheses are made. In each hypothesis, h , the action is taken to be the generalized task action, a , and the goal is a generalization of x_{CGB} . The number of hypotheses expanded is dependent on the size of $crit_{CGB}$. Each expanded hypothesis has a single goal belief x , where $expt = expt_{CGB}$, and $crit$ is some combination of the features in $crit_{CGB}$. For example, if $crit_{CGB}$ has four features, one hypothesis will be the generalized task action and a goal belief with all four features (the most specific hypothesis). Another hypothesis will be the generalized task action and a goal belief with three of the four features, and so on. This expansion results in a hypothesis space of all task representations that are

consistent with the current example of the task. This is illustrated in Fig. 3-3.

The current best representation (the primary hypothesis) is chosen with a Bayesian likelihood method: $P(h|D) \propto P(D|h)P(h)$. The data, D , is the set of all examples seen for this task. $P(D|h)$ is the percentage of the examples in which the state change seen in the example is consistent with the goal representation in h . For priors, $P(h)$, the system prefers a more specific hypothesis over a more general one (as determined by the number of goal beliefs, and number of criteria and expectation features in those beliefs). Thus, when a task is first learned, every hypothesis is equally represented in the data, and the system chooses the most specific hypothesis for the next execution.

3.2.4 Execution of a Known Task

If Leo is asked to do a task that he already knows, T_{known} , he first checks to see if the goal, G , is complete: $\forall x \in G$, if any belief $b \in B$ (of the Belief System) matches all of the $crit \in x$, then b must also match all of the $expt \in x$.

If this does not hold true for any $b \in B$, then the Task Learning Module uses the primary hypothesis of T_{known} to achieve the task. Each of the executables $(e, G) \in E$ is put on a stack. The system executes each e_i to achieve the associated G_i . If e_i is a task then its executables are pushed onto the stack. If e_i is a generalized task then its executable is the name of the primitive action, a to be applied to any beliefs not meeting the goal. For every belief $b \in B$ that matches the $crit \in x \in G$ but not the $expt \in x \in G$, the system puts an action, a with object of attention b , onto the stack.

Leo is persistent about the goals of executables. Occasionally, an action will fail to have the desired effect and in this case Leo will repeat the executable e_i to bring about G_i before moving on.

The primary hypothesis used for execution has a likelihood (between 0 and 1) relative to the other hypotheses available. If this likelihood is low ($< .5$), Leo expresses tentativeness (frequently looking between the instructor and an action's object of attention). Upon finishing the task, Leo leans forward with his ears perked waiting for feedback. The teacher can give positive verbal feedback (e.g., "Good", "Good job", "Well done", ...) and Leo considers the task complete. When the task completes the hypothesis space is updated: $\forall h \in T_{known}$ (including the h used for execution) if the actions and state changes of this most recent demonstration are consistent with h then $f = f + 1$. Thus, $P(D|h)$ for these hypotheses increases in our Bayesian likelihood calculation, relative to the hypotheses not consistent with this example. The primary hypothesis remains the same as it will still be the most specific.

After completing the demonstration, if Leo has not yet achieved the goal, the human can give negative verbal feedback (e.g., "No", "Not quite", ...) and Leo goes back

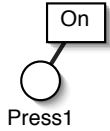
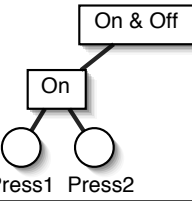
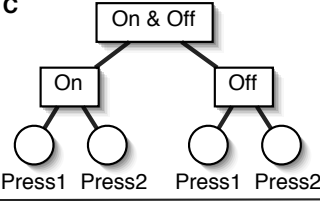
Learning Dialog	Progressive Task Representation	
<p>"Can you do Buttons On&Off?" Leo shakes head no, ready to learn</p> <p>"First, do Buttons On" Looks confused, needs to learn this too</p> <p>"Press Button 1" Presses, sees the state change (see Task rep: A)</p> <p>...same is done for Button 2</p> <p>"Now, Buttons On is done" Confirming nod, sees state change Saves subtask to task set Continues original task (see Task rep: B)</p> <p>....same is done for Buttons Off subtask</p> <p>"Leo, Buttons On&Off is done!" Gives confirming head nod Notices no state change for task Goal = just-do-it (see Task rep: C)</p>	A	 <p>Action: Press Button 1 Goal: $\text{expt}\{\text{On}\}$ $\text{crit}\{\text{Obj}, \text{Button}, \text{name}=1, \text{loc}=\dots\}$</p>
	B	 <p>Task: ButtonsOn Actions: Press B1, Press B2 Goal: for B1 and B2 $\text{expt}\{\text{On}\}$ $\text{crit}\{\text{Obj}, \text{Button}, \text{name}=\dots, \text{loc}=\dots\}$</p>
	C	 <p>Task: ButtonsOn&Off Actions: ButtonsOn, ButtonsOff Goal: just-do-it</p>

Figure 3-4: Learning to turn two buttons ON and OFF, and the progressive task and goal representation. Initially there are two buttons in front of Leo, Button1 and Button2, and they are both in the OFF state.

into learning mode and expects the teacher to lead him through the completion of the task. In this refinement stage, a new hypothesis h_{new} is created. This h_{new} contains the executables of the primary hypothesis which Leo completed on his own, and additional executables that are added as the human requests refinement actions. The goal of h_{new} is inferred once the human indicates the task is complete. A space of hypotheses consistent with this refined example is expanded, as described in the previous section. For each of these, if it already exists in T_{known} then its f is incremented, otherwise it is added (with $f = 1$). Again, the primary hypothesis of T_{known} is chosen with the Bayesian likelihood method.

3.2.5 Example Learning Results

In the test scenario, there are various buttons of different colors in front of Leonardo. The buttons can be pressed ON or OFF (switching an LED on or off). The robot is able to learn several tasks in this scenario of both simple and complex hierarchies, and has tasks with both state-change and just-do-it goals (e.g. turning a set of buttons ON or OFF, and turning a button ON and OFF as a separate task or as a sub-task of a larger sequence). The robot is able to recall tasks learned as sub-tasks of larger tasks as well as correctly associate state-change goals and just-do-it goals.

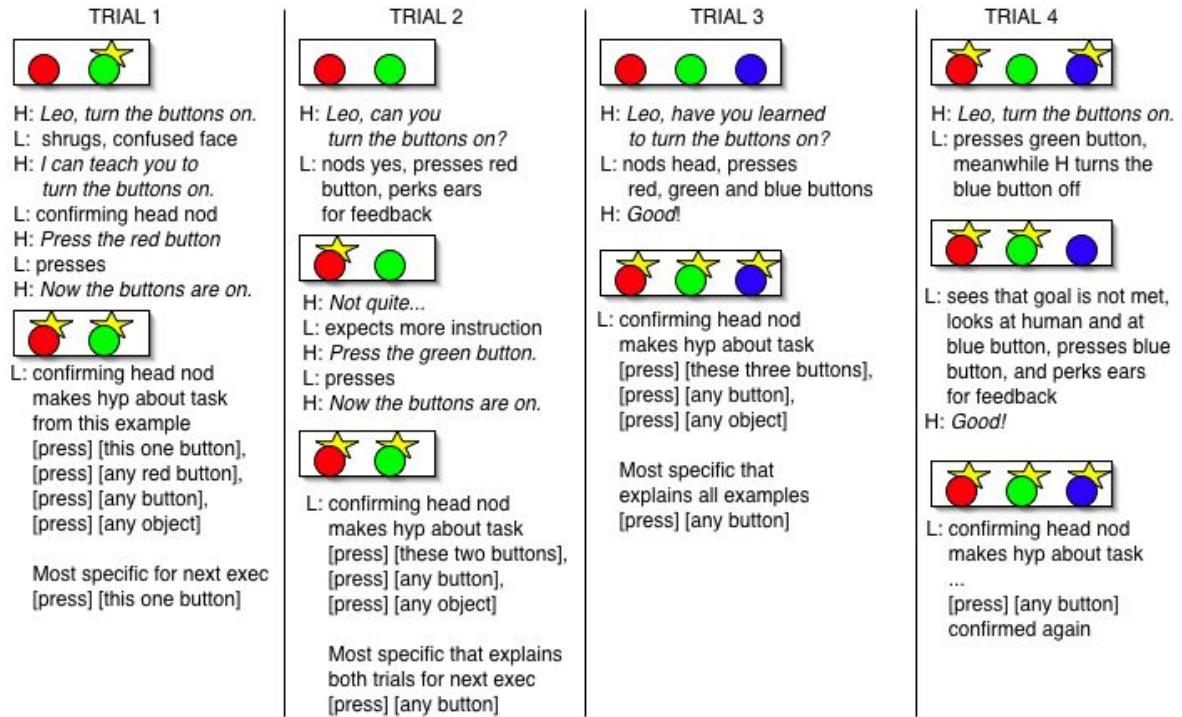


Figure 3-5: Four trials of an interaction in which a human (H) teaches Leo (L) to “Turn the buttons ON.” From left to right the buttons are red, green, and blue. An ON button is indicated with a star, OFF does not have the star.

As one example, Figure 3-4 shows how the task and goal representation develops throughout an interaction with the human partner as they teach Leo to turn two buttons ON and then OFF. This task has both state-change and just-do-it goals, and the subtasks are learned within the larger task. Initially the human is in front of Leonardo and there are two buttons (labeled Button 1 and Button 2), both are in the OFF state. The human asks Leo to “Do Buttons On & Off,” to which Leo shrugs to indicate he does not know and they do the “I can teach you” exchange. Then Leo is in learning mode, and the human asks him to “Do Buttons On.” Again Leo does not know, shrugs, and begins to learn this subtask. The human asks Leo to “Press Button 1.” Doing so, Leo infers the state-change goal for this action. The same happens for “Press Button 2,” and then the human says “Now Buttons On is done.” This causes Leo to: 1) infer a goal (with two goal beliefs) of the entire Buttons On task; 2) add Buttons On to *Tasks*; and 3) return to learning Buttons On&Off adding Buttons On as an executable. The Buttons Off subtask is learned in a similar fashion, and finally the human says, “Leo, Buttons On&Off is done!” When Leo infers a goal for the entire task, he sees that there is no state change and considers it a just-do-it goal.

As a second example, Figure 3-5 shows a transcript from a session in which a human

teaches Leo to “Turn the buttons ON.” The initial trial starts with two buttons visible and the green button already on. The human asks Leo to press the red button to make both buttons ON. This produced four hypotheses about the actual task representation, and the most specific is chosen for the next execution of “Turn the buttons ON.” In the second trial, the teacher structures the task (starting with both buttons OFF) to resolve an ambiguity from the previous example, giving Leo another key example of “Turn the buttons ON.” Following this example, three hypotheses explain the two examples seen thus far, and the most specific is to “press any button.” Therefore, Leo exhibits the correct behavior in trial 3. In trial 4 the teacher tests Leo’s understanding of the overall goal, and Leo shows commitment to the “any button ON” goal. This is an example in a low dimensional feature space with relatively few ambiguities to resolve, but nevertheless demonstrates the advantage of the social dialog paradigm. The human and the robot participate in a tightly coupled interaction in which the human teacher structures the learning process, based on feedback from the robot, such that the robot quickly acquires the examples needed to generalize to the correct goal-oriented task representation.

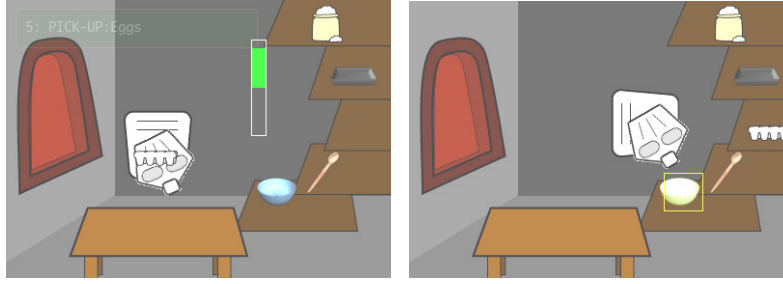
3.3 Using Guidance in Sophie’s Kitchen

Having explored the guidance end of the spectrum, *Sophie’s Kitchen* allows for the investigation of the exploration side of the spectrum. The original version of *Sophie’s Kitchen*, used in Chapter 2, is the extreme of the exploration dimension, allowing for only a limited interaction with a human teacher. The second mechanism of this chapter is a modification of the *Sophie’s Kitchen* game to incorporate more explicit guidance from a human partner.

3.3.1 Modifications to Leverage Human Guidance

The findings in Chapter 2 suggest that people want to speak directly to the action selection part of the algorithm to influence and guide the exploration strategy. To distinguish this intention from feedback, a guidance channel of communication was added. Clicking the right mouse button draws an outline of a yellow square. When the yellow square is administered on top of an object, this communicates a guidance message to the learning agent and the content of the message is the object. Figure 3-6(b) shows the player guiding Sophie to pay attention to the bowl. Note, the left mouse button still allows the player to give feedback as described in Section 2.1.3, but there are no longer object rewards.

Conceptually, the modifications to incorporate guidance give the algorithm a pre-



(a)

(b)

Figure 3-6: The embellished communication channel includes the feedback messages as well as guidance messages. In 3-6(a), feedback is given by left-clicking and dragging the mouse up to make a green box (positive) and down for red (negative). In 3-6(b), guidance is given by right-clicking on an object of attention, selecting it with the yellow square.

Algorithm 2 Interactive Q-Learning modified to incorporate interactive human guidance in addition to feedback.

```

1: while learning do
2:   while waiting for guidance do
3:     if receive human guidance message then
4:        $g = \text{guide-object}$ 
5:     end if
6:   end while
7:   if received guidance then
8:      $a = \text{random selection of actions containing } g$ 
9:   else
10:     $a = \text{random selection weighted by } Q[s, a] \text{ values}$ 
11:  end if
12:  execute  $a$ , and transition to  $s'$ 
    (small delay to allow for human reward)
13:  sense reward,  $r$ 
14:  update Q-value:

```

$$Q[s, a] \leftarrow Q[s, a] + \alpha(r + \gamma(\max_{a'} Q[s', a']) - Q[s, a])$$

```

15: end while

```

action and post-action phase. In the pre-action phase the agent registers guidance communication to bias action selection, and in the post-action phase the agent uses the reward channel in the standard way to evaluate that action and update the Q-value. The modified learning process is shown in Algorithm 2.

The agent begins each iteration of the learning loop by pausing to allow the teacher time to administer guidance (1.5 seconds). The agent saves the object of the human's guidance messages as g . During the action selection step, the default behavior chooses randomly between the set of actions with the highest Q-values, within a bound β . However, if any guidance messages were received, the agent will *instead* choose randomly between the set of actions that have to do with the object g . In this way the human's guidance messages bias the action selection mechanism, narrowing the set of actions the agent considers.

3.3.2 Evaluation

Expert Data

To evaluate the potential effects of guidance, a single expert⁴ completed a series of training sessions, in two conditions:

1. No guidance: has feedback only and the trainer gives one positive or negative reward after every action.
2. Guidance: has both guidance and feedback available; the trainer uses the same feedback behavior and also guides to the desired object at every opportunity.

One user followed the above expert protocol for 10 training sessions in each condition (results in Table 3.1). For the user's benefit, the task was limited for this testing (e.g., taking out the spoon/stirring step, among other things).

The guidance condition is faster: The number of training trials needed to learn the task was significantly less, 30%; as was the number actions needed to learn the task, 39% less. In the guidance condition the number of unique states visited was significantly less, 40%; thus the task was learned more efficiently. And finally the guidance condition provided a more successful training experience. The number of trials ending in failure was 48% less, and the number of failed trials before the first successful trial was 45% less.

Non-Expert Data

Prior works have pointed out how supervision or guidance might benefit a machine learner [Clouse and Utgoff, 1992, Smart and Kaelbling, 2002], and the expert experi-

⁴the author

Table 3.1: An **expert** user trained 20 agents, with and without guidance, following a strict best-case protocol in each condition; this yields theoretical best-case effects of guidance on learning performance. (F = failed trials, G = first success). The following are the results of 1-tailed t-tests.

Measure	Mean no guide	Mean guide	chg	t(18)	p
# trials	6.4	4.5	30%	2.48	.01
# actions	151.5	92.6	39%	4.9	<.01
# F	4.4	2.3	48%	2.65	<.01
# F before G	4.2	2.3	45%	2.37	.01
# states	43.5	25.9	40%	6.27	<.01

ment verifies that guidance has the potential to drastically improve several metrics of the agent’s learning performance. However, the primary interest and contribution of this work is the focus on ordinary human teachers. Thus, the final evaluation looks at how the agent performs when ordinary human trainers are able to provide guidance and attention direction.

Additional people were solicited to play the *Sophie’s Kitchen* game using both feedback and guidance messages. The following instructions about the guidance messages were added to the instructions from the previous experiment (and mentions of object specific rewards were removed).⁵⁶

You can direct Sophie’s attention to particular objects with guidance messages. Click the right mouse button to make a yellow square, and use it to help guide Sophie to objects, as in “Pay attention to this!”

The game logs of these players (the guidance condition) are compared to a second group who played with feedback only, without the guidance signal (the no guidance condition). This comparison is summarized in Table 3.2.

Guidance players were faster than no guidance players. The number of training trials needed to learn the task was 48.8% less, and the number actions needed was 54.9% less. Thus, the human teachers were able to guide the agent’s attention to appropriate objects at appropriate times to create a significantly faster learning interaction.

The guidance condition provided a significantly more successful training experience. The number of trials ending in failure was 37.5% less, and the number of failed trials before the first successful trial was 41.2% less. A more successful training experience is particularly desirable when the learning agent is a robot that may not be able

⁵The full protocol, instructions and consent form used in the study can be found in Appendix A.

⁶We had both male and female participants, but did not keep gender statistics of the population.

Table 3.2: Non-expert human players trained Sophie with and without guidance communication available and also show positive effects of guidance on the learning performance. (F = failed trials, G = first success). The following are the results of 1-tailed t-tests.

Measure	Mean no guide	Mean guide	chg	t(26)	p
# trials	28.52	14.6	49%	2.68	<.01
# actions	816.44	368	55%	2.91	<.01
# F	18.89	11.8	38%	2.61	<.01
# F before G	18.7	11	41%	2.82	<.01
# states	124.44	62.7	50%	5.64	<.001
% good states	60.3	72.4		-5.02	<.001

to withstand very many failure conditions. Additionally, a successful interaction, especially reaching the first successful attempt sooner, may help the human teacher feel that progress is being made and prolong their engagement in the process.

Finally, agents in the guidance condition learned the task by visiting a significantly smaller number of unique states, 49.6% less than the no guidance condition. Moreover, we analyze the percentage of time spent in a good portion of the state space, defined as $Good = \{\text{every unique state in } X\}$, where $X = \{\text{all non-cyclic state sequences, } \{s_0, \dots, s_n\}, \text{ such that } n \leq 1.25(\text{min_sequence_length}), \text{ and } s_n = \text{a goal state}\}$. The average percentage of time that guidance players spent in $Good$ was 72.4%, and is significantly higher than the 60.3% average of no guidance players. Thus, attention direction helps the human teacher keep the exploration of the agent within a smaller and more positive (useful) portion of the state space. This is a particularly important result since that the ability to deal with large state spaces has long been a criticism of RL. A human partner may help the algorithm overcome this challenge.

3.4 Socially Guided Exploration

Leonardo’s ability to learn within a social dialog exhibits several qualities that are desirable for a SG-ML system.

- Learning happens within a tightly coupled interaction, where the robot’s demonstrations of the hypothesized task representations are able to help the instructor pick the seminal examples still needed.
- Nonverbal social cues frame the interaction, establishing mutual beliefs about the state of the task and the state of the robot’s attention.

- Learning is goal-oriented and assumes that the human partner is communicating in goal-oriented ways.
- Leonardo incorporates feedback from the human partner to quickly refine the representation of a task goal.

The Social Dialog system is positioned on the guidance end of the guidance-exploration spectrum. On the opposite end of the spectrum, Sophie’s self-exploration also exhibits several desirable qualities for a SG-ML system.

- Often a teacher gives a learner general guidance while the learner explores the space of a task. (e.g., Imagine teaching someone to ride a bicycle, it is easier to give high level feedback rather than precise instructions about the movement.) One benefit of an exploratory learner is that the teacher need not know exactly what the learner needs to do to complete the task.
- Any realistic learning scenario for an SG-ML system will require that it be able to learn and explore on its own when a human teacher is not available. Thus a second benefit of self-exploration is that it does not require the human’s presence or undivided attention in order for learning to take place.

Having experimented at both ends of the guidance-exploration spectrum, it becomes clear that a social learner cannot simply occupy a single point on this scale, they must have both capabilities. An ideal SG-ML system is able to learn on its own through exploration, but also seamlessly incorporate the guidance of a human partner. The final learning mechanism implemented on the Leonardo platform, Guided Exploration, is motivated to learn and explore the environment but also has the ability to take advantage of social structure provided by a human teacher.

3.4.1 Foundations for Self-Motivated Exploration

In creating a Guided Exploration learning mechanism for Leonardo, the first step is self-motivated behavior and exploration. Note that previous versions of the Leonardo behavior system have not been proactive. For instance, in the Social Dialog learning scenario, the robot continually awaits instruction from the human partner.

Recently there have been a few related works in the realm of internal motivations for a reinforcement learner. Intelligent adaptive curiosity is an approach that uses a *progress drive*, where learning progress is defined as the error in the prediction model, $P(s_{t+1}|s_t, a)$ [Oudeyer and Kaplan, 2004]. In essence, the agent is ‘motivated’ to learn

the world completely as the reward signal is defined by the agent’s world knowledge. Intrinsically motivated reinforcement learning uses intrinsic motivation in combination with extrinsic environmental rewards. In this case, intrinsic reward is proportional to the novelty of a state transition: $(1 - P(s_{t+1}|s_t))$ [Singh et al., 2005]. New ‘skills’ or options are learned via Q-learning whereby the reward is the combination of the intrinsic reward and any extrinsic reward from the environment. Thus, a novel state change initially increases the reward received after that state change and this diminishes over time until the reward is only the extrinsic reward from the environment. [Ahn and Picard, 2006] have some recent initial work on using emotional models as intrinsic drives for a reinforcement learner. In their implementation, one emotion circuit for ‘wanting’ is used as intrinsic reward in addition to extrinsic environmental rewards.

The primary difference in the approach here is that Leonardo’s motivational drives are not directly influencing the reward signal or value function. In prior works, the internal motivation (particularly some measure of certainty) contributes to the reward signal and thus influences the value function. Thus an action that leads to novelty is positively reinforced to encourage more focus on that portion of the state-action space. In Leonardo, on the other hand, the motivational drives trigger different learning behaviors, but do not contribute to the reward signal used to learn a particular task. For instance, a similar measure of novelty is used as a motivational drive, but rather than directly influencing the value of the state action pair that caused it, the drive triggers the creation of an option to learn more about that state change and how to bring it about.

This section describes several aspects of the internal motivations implemented to create Leo’s self-motivated exploration behavior, and Section 3.4.3 explains how these influence Leo’s behavior to create learning opportunities. Sections 3.4.4, 3.4.5 and 3.4.6 detail how the system takes advantage of these in an options learning framework.

Short Term Memory

The system maintains an event history of *actions* and *states*. Recall from Section 3.2.2 that, s_t is a set of belief objects that contain the values that every percept had at the particular time t . Leo saves the past 100 events $a \in A$ and s_t . A new s_t is added to the event list at times when something about the state has changed. The Short Term Memory structure also builds a transition model, $P(s_2|a, s_1)$, keeping track of the probability that action, a , in state s_1 , will lead to state s_2 .

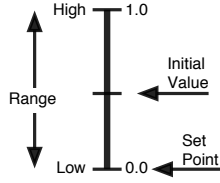


Figure 3-7: Each of Leo's motivational drives has an initial value and a specified range. Within this range it has a set point (the value that it drifts towards).

Motivation System

In living systems, there are certain critical features that must be kept within a bounded range (e.g., amount of food, water, temperature, ...). The process of behavioral responses to maintain acceptable values of these critical parameters is known as homeostatic regulation or behavioral homeostasis [Plutchik, 1984]. If the parameter falls out of the desired range, the animal will become motivated to behave in a way that brings the parameter back into the desired range. In a simplified view these critical parameters can be thought to encode the innate needs of the system.

Leonardo's Motivation System defines its *needs* and how it will act to satisfy those needs (this is based on the Motivation System of the Kismet robot [Breazeal, 2002]). In this case, Leonardo's motivations are designed to guide behavior in a learning mechanism. Inspired by natural learning systems that are driven to learn new things, Leonardo's Motivation System implements three motivational drives meant to produce a learning behavior that a human partner may find natural and understandable.

Drives are implemented as variables which have an initial value and a specified range. Within this range they have a set point (the value that they drift towards), and a drift magnitude (the maximum value they can drift in one clock cycle). All of the motivational drives have a range [0, 1], initial value of 0.5, set point of 0.0, and a drift magnitude of 0.001 (Fig. 3-7). Each clock cycle the Motivation System updates the following drives based on perceptions of internal and external state: Mastery, Novelty, and Activity.

The Activity Drive is meant to reflect the current level of activity. Each cycle that Leo is performing any action, the activity drive drifts toward its maximum value, 1.0; at any other time the drive drifts back toward its set point, 0.

The Novelty Drive is meant to reflect a measure of how novel recent events have been. Each cycle the Motivation System gets the time of the last state change and the degree of the last change, d_{chg} , from Short Term Memory. The degree of a state change is related to the number of times this state change has been seen by the system:

$$d_{chg}(s_1, s_2) = \frac{1}{2 \times frequency(s_1, s_2)}$$

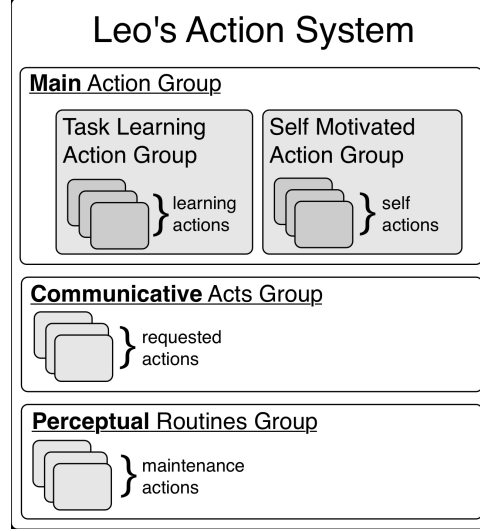


Figure 3-8: Leonardo's Action System has several Actions and Action Groups that compete for control of the behavior at any given time. For the purpose of this thesis the primary focus is the Task Learning Action Group. This group becomes relevant (triggers) in several learning contexts and utilizes various specific actions in these contexts, described in Sec. 3.4.3

Each state change causes the novelty drive to drift towards its maximum value, 1.0, for a period of time, t_{nov} ; the maximum effect on novelty, t_{max} , is 30 seconds.

$$t_{nov} = d_{chg}(s_1, s_2) t_{max}$$

The Mastery Drive is a measure of the level of confidence the system has in the current state. Each cycle this is calculated based on the task set in the Learning Action Group. Mastery is taken as the average confidence of the $T \in Tasks$ relevant in the current state, s . Thus if no tasks are relevant the current level of mastery is 0. A particular task, T , is relevant if it can be initiated from s . Each task representation has a confidence measure: a ratio of the number of successful attempts to the total number of attempts made at this task.

3.4.2 Action System Overview

Leonardo's Action System has several Actions and Action Groups that compete for the control of behavior at any given time. For more implementation details and perspective of the overall behavior system architecture see [Blumberg et al., 2002]. The implementation details of Leo's Guided Exploration concern mainly the Action System, represented in Figure 3-8. For the purpose of this thesis the primary focus is the Task Learning Action Group. This section describes the constructs necessary to understand these details.

In the C5M architecture, a creature has a single Action System that has a set of all the Actions available to the creature. Each action in the Action System is represented in the form $[trig, act, until]$: *act* = the action itself, *trig* = the triggering environmental context for this action, *until* = the context in which the action should terminate once it is running. The representation is hierarchical such that an action, *act*, can be a single behavior or it can be a group of actions. An Action Group triggers in the same way as a primitive action, and upon activation it has some means of determining which of its sub-actions should become active. Thus, the Action System continually activates and deactivates its various Actions (which may be Action Groups). In a particular time step, if the active action has completed, the system chooses probabilistically between all of the actions for which their triggering context is true in the current state.

Most of the SG-ML learning behavior is brought about in the Main Action Group. This is a group in which the sub-actions are mutually exclusive, and each cycle of execution, the current action to run is selected probabilistically weighted by their relative values. The Main group has two sub-actions (both of which are Action Groups), the Task Learning Action Group and the Self Motivated Action Group. In this implementation, learning is given an order of magnitude more value than random self-motivated action.

There are two Action Groups in addition to the Main Action Group. The Communicative Acts Action Group contains action tuples related to human-directed action. For each primitive action Leo is able to do, $\forall a \in A$, the Communicative Acts group contains a tuple whose trigger is the speech parse requesting the action, possibly with an object of attention indicated as well, and whose action is *a*. For example, the speech “Leo, Press Button 1”, triggers the action $a = press[Button1]$. This Communicative Acts group is implemented as a separate Action Group to ensure that the human partner’s requests will be dealt with promptly, rather than arbitrated alongside self-directed action. Thus, when a human is present Leo is very responsive and attentive to their direct commands (e.g., “Leo, do X”). As described below, the Task Learning Action Group also allows for a more subtle action suggestion from the human partner which does not cause an interrupt in the same way as a commanded action. The Perceptual Routines Action Group contains actions related to low-level maintenance of perception. For instance, it has actions that activate to track the human partner and their pointing gestures.

3.4.3 Task Learning Action Group

The focus of this section is on the Task Learning Action Group box of Figure 3-8. Socially guided exploratory learning is implemented as a behavior group that responds to various contexts of Leo’s internal (motivational) and external (social) world with a series of learning behaviors.

For continuity, the notation from Section 3.2 will be used here where possible: A is Leo's primitive actions, $G = \{x_1, \dots, x_y\}$ is a goal representation where each $x \in G$ is a goal belief, and s_t is a set of belief objects that contain the values that every percept had at the particular time t . Let $Tasks = \{T_1 \dots T_j\}$ continue to be the set of known tasks; however, the representation of each $T \in Tasks$ for Guided Exploration is significantly different and will be detailed in Section 3.4.4.

Learning Contexts

Learning actions become active for various reasons, the following nine contexts will trigger the Task Learning Action Group. Many of the triggering contexts are threshold values of one of the motivational drives, in these cases the exact choice of the threshold value was determined empirically as a value that works well in practice to represent “Low” or “High” for the drive.

1. **Novelty High:** The Novelty drive is ≥ 0.95 .
2. **Novelty Low:** The Novelty drive is ≤ 0.1 .
3. **Novelty Positive Change:** This context is active any time the Novelty drive makes a positive change with at least a 0.1 magnitude, it remains active until there is a negative change.
4. **Novelty Negative Change:** Similar to the above context, this is active any time the Novelty drive makes a negative change with at least 0.25 magnitude and is active until there is positive change.
5. **Activity Low:** The Activity drive is ≤ 0.2 .
6. **Mastery High:** The Mastery drive is ≥ 0.5 .
7. **Learn Now:** This context is active when the speech recognition system parses one of several utterances that corresponds to the human labeling a state change. For example, “Look Leo, it’s TaskName-X.”
8. **Suggest Action:** This context is active when the speech recognition system parses one of several utterances that corresponds to the human making a suggestion for an action Leo should do. For example, “Leo, try to Action-X the Object-Y.”
9. **Task Relevant:** The final learning context is when a $T \in Tasks$ is relevant in the current state. The Task Learning Group continually keeps track of how long each of the tasks $T \in Tasks$ has been relevant using a set $C : \forall c \in C, c_i =$ the number of time steps T_i has been relevant; c_i is reset to 0 in the time step that T_i is no longer relevant. The overall relevance measure, R , for any particular time step is the maximum c_i in C . The Task Relevant context becomes active when $R \geq .75$, thus when any task has been relevant for a few seconds.

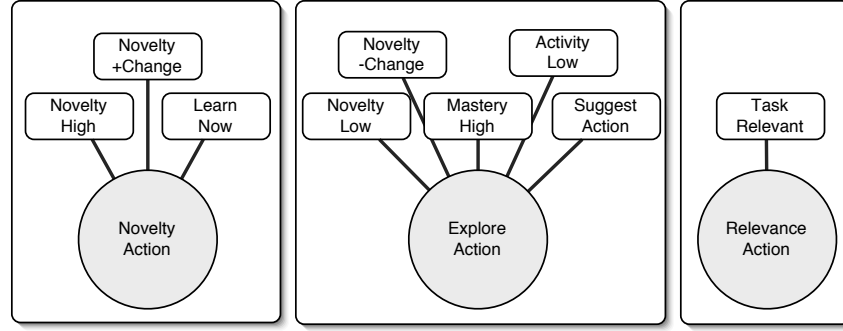


Figure 3-9: The Task Learning Action Group has three competing actions, this figure shows the nine learning contexts in which each action is available.

Learning Actions

The Task Learning Action Group can become active due to any of the nine contexts. Upon activation, the group activates a specific sub-action based on the triggering context. Note that the learning contexts are not mutually exclusive, several are often relevant at once. In this case, the Task Learning Action Group chooses probabilistically between the learning actions that it could activate, this choice is weighted by each action's inherent value. Figure 3-9 illustrates the actions and their associated trigger contexts; and Figure 3-10 illustrates the logic of each learning action.

Novelty action — If the triggering context is Novelty High, Novelty Positive Change, or Learn Now, the Novelty action may be activated. This action has the highest inherent value. This action first gets the most recent state transition (s_1, a, s_2) from the Short Term Memory. Then it makes a goal representation of the change $s_1 \rightarrow s_2$. If this goal is not currently represented by any $T \in Tasks$ then a T_{new} is created for this goal. If a human partner named the task, it is labeled with that name. Then T_{new} is incorporated into $Tasks$. Details of the goal representation, task creation, generalization, and expansion processes are found in Sections 3.4.4 and 3.4.5.

Relevance action — If the triggering context is Task Relevant, the Relevance action may be activated. This action has the lowest inherent value. This action selects randomly from the set of tasks that are currently relevant, and activates this task. Once active, the task takes over execution and selects actions to reach the goal, this process is detailed in Section 3.4.4. Once the task finishes, control is passed back to this Relevance action. It registers whether or not the task was successful, notes this in the task representation, and requests a happy or sad facial expression to correspond to the success or failure of this attempt.

Explore action — If the triggering context is Novelty Low, Novelty Negative Change, Activity Low, Mastery High, or Suggest Action, the Explore action may be activated. This

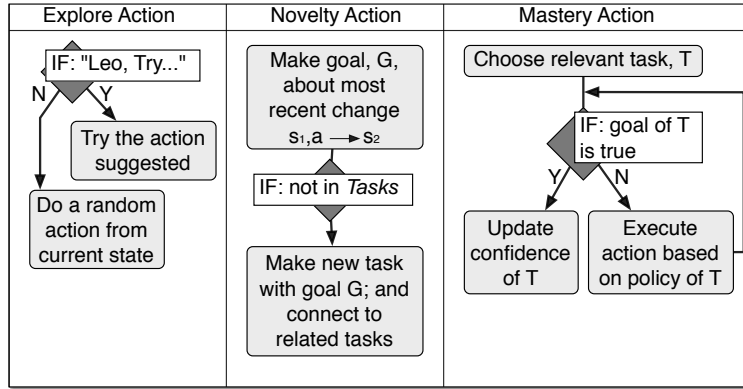


Figure 3-10: The logic executed when each of the three learning actions is triggered.

action has the second highest inherent value. When the explore action is activated, it first checks to see if there was any human-suggested action. If there was, and it is able to do this action, it will. Otherwise, the Explore action will select from the actions it can do in the current state, with a minimum frequency requirement of two. Once the action is completed, if this was a human-suggested action the robot's attention is influenced to try to look up to the human. This acknowledges the suggested action and provides an opportunity for feedback. Whether or not the action was suggested, if after the action the human gave negative feedback, the robot will try to reverse the action. This strategy is discussed further in Chapter 5.

3.4.4 Task Representation

In the same spirit of the Social Dialog learning implementation, this work aims to have a system learn the goal or concept of an object-oriented activity. A goal is a particular state or state change, where a state is a particular time slice of the Belief System. The task or activity representation for Guided Exploration is significantly different than that described in Section 3.2.1 and includes a representation of the goal as well as multiple (context-dependent) ways to achieve this goal.

Csibra's theory of human action serves as inspiration for Leonardo's activity representation, and is consistent with the existing action constructs of the C5M architecture. In the theory, activity has the representation *[context][action][goal]*, and a series of experiments with infants finds that they have efficiency expectations with respect to each of these three [Csibra, 2003]. For instance, given a goal and a context infants expect the most efficient action to be used (and are surprised when it is not); the experiments show the ability to infer goal and context in a similar fashion. In one experiment, 9-12 month old infants were repeatedly shown animations of a ball jumping over an obstacle

to reach and contact a second ball. In this case the jumping action is instrumental to the goal (contacting the second ball). After habituating to this animation the infants are shown the test configuration where the obstacle is gone. In one test condition infants are shown an animation where the approaching ball does the same jumping action to reach the other ball, and in the second test condition the approaching ball makes the more efficient straight-line approach to the other ball. Using looking time as a measure of broken expectations, Csibra found that infants were using a goal-oriented interpretation. Despite habituating to the jumping action, in the test configuration infants preferred the new instrumental straight-line action to the now unnecessary jumping action.

This type of representation is desirable for an SG-ML system because it leads to a reasonable generalization of activity across contexts. For instance, if the system is always trying to build a better model of the *context* component of an activity representation, this will lead to the ability to say, “this looks like the *kind-of-situation* where I do *X*” or abstracted even further “I feel like doing *X*.” Additionally, this representation implies the flexibility to learn multiple ways to accomplish the same goal.

Leonardo’s task representation described in Section 3.2.1 already fulfills several aspects of this activity representation. The contexts, actions, and goals of hierarchical tasks are learned and refined over a few examples. However, the system can only represent one way of achieving each task-goal, and learning was a particular activity rather than a part of all activity. The Guided Exploration version of learning changes a few key aspects of task representation to accommodate the scenario of ‘learning all the time’.

- The human partner is no longer providing distinct start and stop points for the representation task, the robot decides that a particular state change is interesting and creates a task representation to learn how to bring this state about (Sec. 3.4.5).
- Once a task representation is created, all of the robot’s actions can be learning opportunities. Even when a particular task is not actively being explored any experience can update the policy of this task as if it were the current goal (Sec. 3.4.5).
- The action representation portion of the task is a policy of action, which assumes there may be multiple ways to achieve a goal depending on the state of the world.

The system uses Task Option Policies for this more flexible task representation. This name is chosen to reflect the similarities to the Options framework in the Reinforcement Learning literature [Sutton et al., 1999]. Options are made up of three constructs (I, π, β) , where S is the state space and A is the action space:

- $\pi : S \times A \rightarrow [0, 1]$; A policy estimating a value for (s, a) pairs.
- $\beta : S^+ \rightarrow [0, 1]$, where $S^+ \subset S$; is all the states in which this option terminates.
- $I \subseteq S$; is all the states in which this option can initiate.

An option can be taken from any state in I , then actions are selected according to π until the option terminates stochastically according to β .

A Task Option Policy, $T \in Tasks$, is defined by very similar constructs (I', π', β') . Let $S_{task} \subset S$ be the subset of states in which the task is relevant but not yet achieved, and $S_{goal} \subset S$ be the subset of states in which the task goal is achieved.

- $\pi' : S_{task} \times A \rightarrow [0, 1]$; estimates a value for (s, a) pairs for achieving the task goal.
- $\beta' : S_{goal}$; represents all of the states in which this task terminates because the task goal, G , is true.
- $I' = S_{task}$. The task can be initiated in all of the states relevant to the task, for which the task has a policy of action.

Thus, a task, T , can be taken (i.e., the Task Relevant learning context is true) when the current state is one of the states S_{task} , then actions are chosen according to π' until the current state is one in S_{goal} in which $G \in T$ is true (with some probability of terminating before G is true. i.e., giving up). Recall from Sect. 3.2.4 that goal completion is tested by the following: $\forall x \in G$, if any belief $b \in B$ (of the Belief System) matches all of the $crit \in x$, then b must also match all of the $expt \in x$.

Having defined the Task Option Policy representation, the following two sections detail how Leonardo learns a new Task Option Policy by creating a new goal G and expanding and generalizing the set S_{task} , goal G , and policy π' over time.

3.4.5 Learning Task Option Policies

When the Novelty Action is activated, a potential goal state G is made from the most recent state change, (s_1, a, s_2) . The procedure for making a goal state, G , given two states, s_1 and s_2 is the same as described in Section 3.2.2. If there is not currently a $T \in Tasks$ with the goal G then a new Task Option Policy, T_{new} , is created with the goal state G .

The set S_{task} of T_{new} is initialized with the single initiation state s_1 , and the action policy π' is initialized with default values $q = .1$ for all actions from s_1 . Then the system takes into account the experience of (s_1, a, s_2) , and the pair (s_1, a) is given a higher value since s_2 represents the goal state. The experience and update process is described below. Having created T_{new} , the system adds it to $Tasks$. When it is incorporated into the set it is linked or connected to other related tasks:

- If there is a task $T_i \in Tasks$ that has s_2 in its initiation set S_{tasks} then expand the policy of T_{new} by adding the S_{task} and π' of T_i to the S_{task} and π' of T_{new} .
- Additionally if there is a task $T_i \in Tasks$ for which its goal G_i is true for s_1 , then add the state action pair (s_1, a) of T_{new} to the policy of T_i .

Algorithm 3 With each experience $(s_1, a \rightarrow s_2)$, every task has the opportunity to learn, with the possibility of both extending and updating its policy.

```

1: for each  $T$  in  $Tasks$  do
2:    $G$  = the goal of  $T$ 
3:    $S_{task}$  = the initiation set of  $T$ 
4:   if  $(s_1$  not in  $S_{task})$  AND  $(G$  not true in  $s_1)$  AND
       $((G$  true in  $s_2)$  OR  $(s_2$  is in  $S_{task}))$  then
5:     Extend: add  $s_1$  to  $S_{task}$ 
6:   end if
7:   if  $s_1$  is in  $S_{task}$  then
8:     Update the value of  $[s_1, a]$  in  $\pi'$ :
9:      $r=0$ 
10:    if  $(G$  is true in  $s_2)$  then
11:       $r=1$ 
12:    end if
13:     $Q[s_1, a] \leftarrow Q[s_1, a] + \alpha(r + \gamma(\max_{a'} Q[s_2, a']) - Q[s_1, a])$ 
14:  end if
15: end for

```

Each $T \in Tasks$ has the opportunity to learn and expand from every experience (this is also referred to as off-policy or intra-option learning [Sutton et al., 1998]). Each action the robot takes is an experience, (s_1, a, s_2) . In the case where an action does not have an effect, $s_1 = s_2$. Each $T \in Tasks$ is given the opportunity to extend its set S_{task} and update its policy π' based on this experience (also shown in Algorithm 3):

- Extend: $\forall T_i \in Tasks$, if $s_1 \ni S_{task}$ of T_i and G_i is not true for s_1 , then include s_1 in the S_{task} of T_i if and only if G_i is true for s_2 or $s_2 \in S_{task}$ of T_i .
- Update: $\forall T_i \in Tasks$, if $s_1 \in S_{task}$ then update the value of (s_1, a) in the π' of T_i : $Q[s_1, a] = Q[s_1, a] + \alpha(r + \gamma \max_a (Q[s_2, a]) - Q[s_1, a])$, where $r = 1$ if and only if goal G_i of T_i is true in s_2 , otherwise $r = 0$.

Any Task Option Policy, T , is considered relevant if the current state s is in the S_{task} of T . Relevance is the only precondition for activating a task. When T is activated it selects actions based on its policy, π' , selecting the action a that has the highest value from state s . When the goal state is reached, T deactivates, and there is a 10% probability of deactivating after each action that does not end in the goal state. It is important to have some probability of ending the task before it completes, to insure that the agent does not forever attempt a task goal that is perhaps no longer able to be achieved. This 10% probability of “giving up” is arbitrarily chosen and remains constant. In future work it would be interesting to have this probability be dynamic and based on internal motivational states.

Upon deactivation, T updates its confidence measure based on whether or not the attempt was successful. Confidence is simply how many times this task has been successfully completed proportional to how many times it has been attempted.

The primary difference between this approach and others is the goal-oriented nature of the learning. In this case, the novelty drive triggers the creation of a new goal. This trigger can be *influenced* by the human partner (if they label the goal state for example with a statement such as “Look Leo, it’s X”), but the human is not *required* to provide the goals. In defining its own goals the system is framing its own learning problem. Similarly, as these Task Option Policies are developed, the human partner is not required to define a reward signal. The system frames its own learning problem, by assuming that being in the goal state has the highest reward for that particular Task Option Policy and a standard reinforcement learning process works to build a value function for the state action pairs in the vicinity of the goal state.

Often a reinforcement learning agent is meant to learn a model of the world, and learn how to maximize the rewards from the environment. In this approach however, the agent defines goal states for itself, and uses reinforcement learning to build an option representation of how best to achieve that goal from related states. This goal-oriented approach of having a reinforcement learner define what options are good to know, framing its own learning problems, is a novel and important quality of an SG-ML system.

3.4.6 Task Generalization

In this learning mechanism, like the Social Dialog mechanism, generalization is particularly important. The Social Dialog learning mechanism actively expanded and refined a hypothesis space of representations of the examples of a task. The Guided Exploration mechanism has a different strategy. Once a Task Option Policy is created, rather than expand a space of hypotheses, the most specific state representations are used and throughout activity the system uses two specific mechanisms to generalize the application of the task: between-policy generalization and within-policy generalization.

Both of these generalization mechanisms work to generalize the state representations in S_{task} and the goal representation G for all $T \in Tasks$. In doing so these processes expand the portion of the state space in which tasks can be initiated or considered achieved. Referring back to the discussion in Section 3.4.4, this is analogous to refining the *context* and the *goal* aspects of the activity representation.

Between-policy generalization

Given two tasks $T_1 \in \text{Tasks}$ and $T_2 \in \text{Tasks}$ ($T_1 \neq T_2$), the between-policy generalization mechanism determines if it is appropriate to combine them into a more general task T_{gen} . For example, if T_1 has the goal of turning ON a red button in location (1,2,3), and T_2 has the goal of turning ON a red button in location (4,5,6), then a between-policy generalization would create a T_{gen} with the goal of turning ON a red button without any location specification. When a feature is generalized from the goal representation we also try to generalize all of the state representations in S_{task} , thus T_{gen} no longer pays attention to that feature. Therefore, T_{gen} is now able to initiate in any location, and any state that has a red button ON achieves the goal of T_{gen} .

This between-policy generalization is attempted each time a T_{new} is added to Tasks . If there exist two tasks T_1 and T_2 with similar goal states, then the system makes a general version of this task. Similarity is determined in the following way:

- Let G_1 = the goal of T_1 ; G_2 = the goal of T_2 .
- G_1 and G_2 must have the same number of goal beliefs.
- For each goal belief, $x_1 \in G_1$ there must be a goal belief, $x_2 \in G_2$ such that $(expt \in x_1) = (expt \in x_2)$ and $crit \in x_1$ differs from $crit \in x_2$ by no more than four percepts.⁷
- Let D be a set containing all $crit$ percept values that differ between G_1 and G_2 .

Once T_1 and T_2 are determined to have similar goals, a new task T_{gen} is created that removes any features different between the two. The goal G_{gen} is made for T_{gen} , where

$$(expt \in G_{gen}) = (expt \in G_1)$$

$$(crit \in G_{gen}) = (crit \in G_1) \cap D$$

Now T_{gen} has a generalized goal, in a similar fashion the system tries to generalize the S_{task} and π' of T_1 and T_2 .

- Let $S_{gen} = (S_{task} \in T_1) \cap D$
- Let each of the S_{task} sets in T_1 and T_2 be temporarily changed to S_{gen}
- If $(\pi' \in T_1) = (\pi' \in T_2)$ then T_{gen} uses the generalized set S_{gen} and $(\pi' \in T_1)$.
- If $(\pi' \in T_1) \neq (\pi' \in T_2)$ then S_{gen} is not possible for T_{gen} , instead it is made to use the conjunction of the original policies of T_1 and T_2 , thus using both specific ways of achieving this more general version of the task goal:

$$(S_{task} \in T_{gen}) = (S_{task} \in T_1) \cup (S_{task} \in T_2) \text{ and } (\pi' \in T_{gen}) = (\pi' \in T_1) \cup (\pi' \in T_2).$$

⁷Four is somewhat arbitrary, chosen empirically as a good balance between over and under utilization of the generalization mechanism.

Returning to the red button example, the two tasks are considered similar since their expectations are the same, $expt = \{ON\}$, and their criteria differ only by the location feature, $D = \{loc = (1,2,3), loc = (4,5,6)\}$. Thus a new task is made with a goal that does not include location: $G_{gen} = \{expt, crit\}$, $expt = \{ON\}$ and $crit = \{object, red, button, \dots\}$. If the policies of the two tasks are similar, for example to do the press action in the state $s = \{b_1 = \{object, red, button, loc = (x, y, z), \dots\}\}$, then the new task will have a generalized policy that does not include location. On the other hand, if T_1 has the policy of doing the press action in state $s = \{b_1 = \{object, red, button, loc = (1,2,3), \dots\}\}$, and T_2 has the policy of doing the flip action in state $s = \{b_1 = \{object, red, button, loc = (4,5,6), \dots\}\}$, then the generalized task will maintain that in $loc(1,2,3)$ a red button should be pressed to make it ON and in $loc(4,5,6)$ a red button should be flipped to make it on. These simplified examples are illustrated in Figures 3-11(a) and 3-11(b).

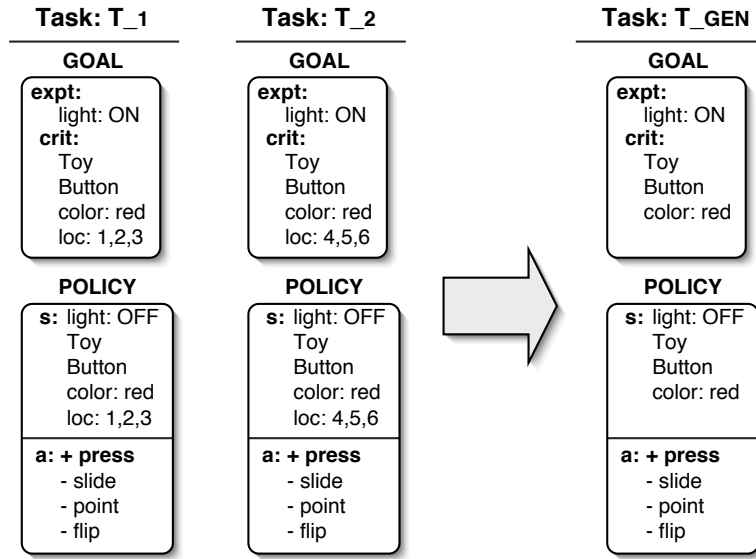
Within-policy generalization

In addition to generalizing between two $T \in Tasks$, it is also possible to occasionally generalize within a task. Within-policy generalization is attempted each time a change is made to the task. For example, recall that every experience tuple (s_1, a, s_2) has the possibility of extending the set S_{task} , each time the set changes the system tries within-policy generalization.

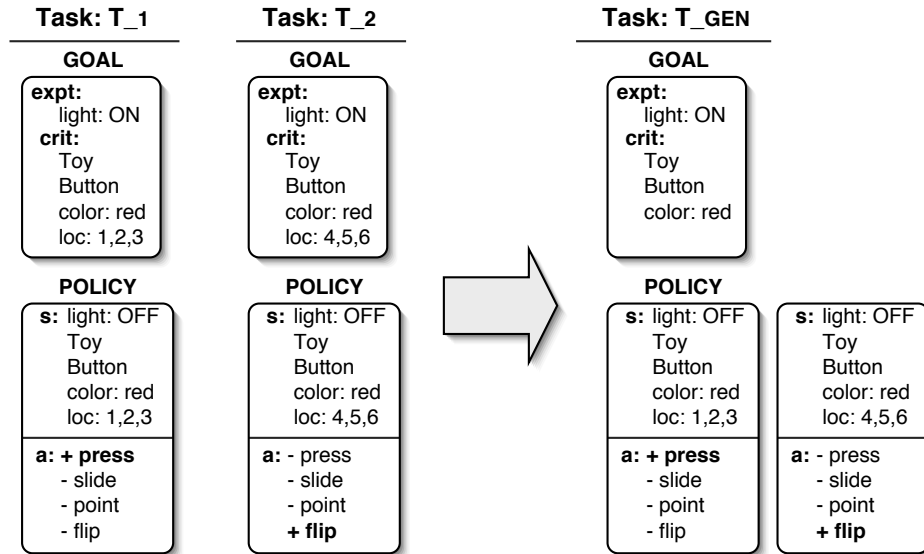
The system tries to find state action pairs in the policy that are similar enough to generalize (i.e., two different states, $s \in S_{task} : s_i$ and s_j , such that the values in π' for s_i and s_j are the same). Thus, since the action policy is the same, the system tries to replace s_i and s_j in S_{task} with a general state s_{gen} that contains all the features they have in common: $s_{gen} = s_i \cap s_j$.

In practice within-policy generalization has the important purpose of allowing for refinement of an over specific between-policy generalization. Consider the example seen in Fig. 3-11(b), where the two tasks were seen to have different action values and thus the generalized policy contains both specific initiation states. Perhaps through later experience and adjustments to the value function, the robot finds that the press action is actually the most valuable action from both of these initiation states. Then this within-policy generalization will work to produce the representation seen in Fig. 3-11(a).

In generalizing the states in S_{task} and the goal representation G for all $T \in Tasks$, these generalization mechanisms expand the portion of the state space in which tasks can be initiated or considered achieved. This makes for a more efficient representation, as the system continually makes the state space representations more compact. Additionally, this works to afford a goal-oriented approach to domain transfer, as the system is continually refining the *context* and the *goal* aspects of the activity representation.



(a) Task T_1 and T_2 have similar goals, to turn the red button ON. So a general task T_{gen} is made with the generalized G , S_{task} , and π' , that no longer include the location feature.



(b) Task T_1 and T_2 have similar goals, to turn the red button ON. So a general task T_{gen} is made with the generalized G . But they have different ways of achieving this goal, so the S_{task} and π' are not generalized, but include the S_{task} and π' from both T_1 and T_2 .

Figure 3-11: Between-policy generalization example: Fig. 3-11(a) shows the generalization for the example where the two tasks have similar goals and action policies. Fig. 3-11(b) shows the example where they have similar goals but different action policies.

3.4.7 Scaffolded Learning

Given the foundation of motivated behavior and mechanisms for goal-oriented learning, the final piece of Guided Exploration involves the mechanisms of social scaffolding that an SG-ML system should be able to leverage. Learning in a social environment is characterized by socially guided discovery, it is the balance between learning on one's own and benefiting from the social environment. To succeed the system needs to be able to explore on its own *and* take advantage of social interaction if it is there. The following are the specific social scaffolding mechanisms at work on the Leonardo platform to enable socially guided exploration and discovery:

- **Social attention:** The attention of the robot is directed in ways that are intuitive for the human. Attention responds to socially salient stimuli and stimuli that are particularly relevant to the current goals of the system. Additionally, the robot tracks pointing gestures and head pose of a human partner which contributes to the saliency of objects and their likelihood of attention direction. For an overview of the robot's social attention abilities see [Thomaz et al., 2005a].
- **Guidance:** Throughout the learning interaction, the human can suggest actions for Leo to try. This is very similar to the Social Dialog version where the human had to instruct Leo about every action. The subtle difference in this Guided Exploration case is that the human's request is treated by the system as a suggestion rather than an interrupt. The suggestion increases the likelihood that the Explore learning context will trigger, but there is still a non-zero probability that Leo will decide to practice a relevant known task or learn about a novel state change.
- **Metrics of success:** The system uses the human partner to help recognize success and failure during learning. The speech recognition grammar contains several phrases that the human partner can use to indicate positive or negative feedback to the robot. If at any point positive or negative feedback is received it is incorporated into the action policy of the current task being executed. Additionally, Leo will occasionally look up to solicit feedback from the human partner when confidence is low or when he has just performed a suggested action.
- **Recognizing goal states:** In the Social Dialog version of learning, the robot was completely dependent on the human to provide the start and end points of task examples. This Guided Exploration version significantly loosens those constraints such that Leo is able to explore on his own and form task representations about novelties in the environment. Additionally, the human can point out goal states with a variety of speech utterances (e.g., "Look Leo, it's X"). This serves to increase

the likelihood that the Novelty learning context will trigger (creating a task representation of this change). The created task is given the label “X” allowing the human refer to it in the future.

- Environmental structure: A key component of social interaction is the actual physical structuring of the environment and the task. The human helps the system proceed at a reasonable learning pace and helps the system notice the big landmarks or important parts of the task. Drawing the system into new generalizations is a large contribution of the human partner, helping to link old information to new situations, pointing out when a learned task is relevant in the current situation.

3.4.8 Example learning results

Leo's Guided Exploration has been developed and tested with a playroom scenario. Given the limited dexterity and perceptual capabilities of the robot, more complex tasks and activities can be learned in simulation with virtual Leo. In simulation, the playroom has several different toy boxes and toy blocks, offering a rich and complex state space. In the real world, Leo's playroom has toy boxes, designed specifically for Leonardo's manipulation capabilities, that can open and close and change color in reaction to various actions. Figure 3.4.8 shows Leo's real and virtual playroom scenarios. All of the learning mechanisms and processes described in the previous sections run in real-time on a dual G5 Macintosh computer.⁸ This section provides some insight into the nature of the tasks both virtual and real Leo are able to learn, and the process of the learning and generalization that occurs.

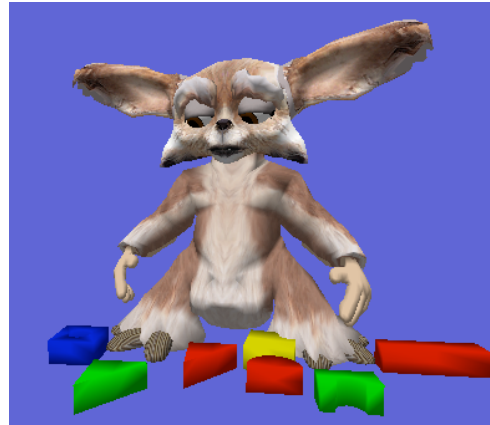
Leo has several primitive arm actions in his repertoire: a pressing down motion, a lifting motion, a sliding motion to the left or right, a hand flip motion, a grasping motion, and a pointing motion. These actions can be directed toward any object in the environment. Leo has no initial knowledge about the objects in the environment, but is able to fully perceive their features. Through self-exploration or guided exploration he is able to build a task set with various goals he is able to bring about in the world.

The objects in the playroom make up a complex state space as a learning environment. This section presents various characterizations of the Guided Exploration learning mechanism. To illustrate its functionality data was collected in several experimental

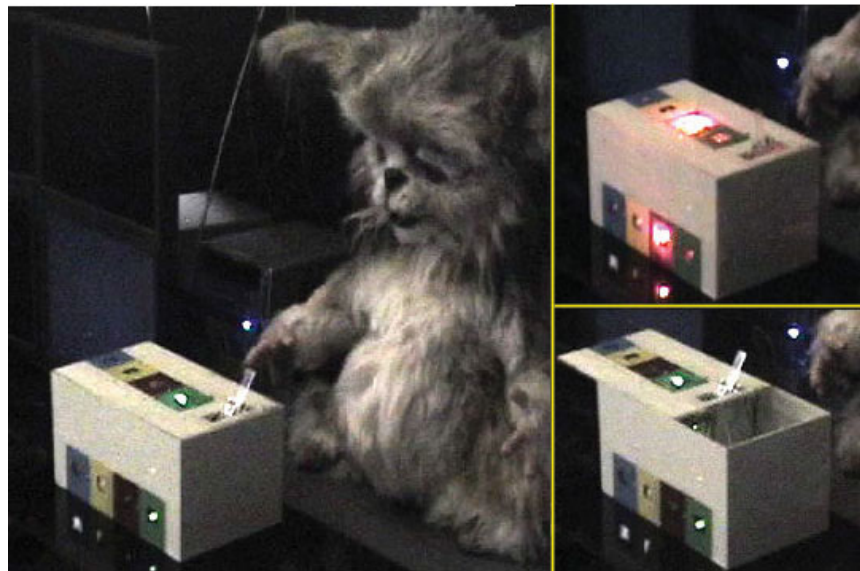
⁸Additional computers are used when Leo is running in the real world: Two Linux machines run processes to grab video from the stereo cameras. Two PCs run computer vision processes to analyze these video streams to recognize people, their headpose, their pointing gestures, and toy objects in the environment. One PC runs the Sphinx speech recognition, and a Mac server runs the motor control interface process. These processes communicate over an internal gigabit network with the IRCP communication protocol described in [Hancher, 2003].



(a) These are two of the five toy boxes Leo has in the virtual world. On the left is a box where pushing the lever flips the lid open. On the right is a different box with a lid that slides open and closed. Both can change colors. Though not graphically pictured, both have a dial that can be turned right or left, and a switch that can be on or off.



(b) There are also various colored blocks from which tasks can be created



(c) In the real world leo has toy boxes that he can change with a gestural interface. The boxes change color, the lid opens, and a physical switch changes state.

Figure 3-12: Leo's playroom, experimental scenarios for Guided Exploration in both the virtual and physical world.

learning session in the virtual playroom. The objects used in the experiment were two toy boxes that have some similarities and some differences in their functionality:

- The Slide-Box: The lid opens with a slide-out action, with the precondition that the switch is ON. The lid closes with the slide-in action, with the precondition that the switch is OFF. The switch turns on and off with the flip action. A dial on the box turns left and right with the squeeze action.
- The Push-Box: The lid opens with a press action, with the precondition that the switch is ON. The lid closes with the press action, with the precondition that the switch is OFF. The switch turns on and off with the flip action. A dial on the box turns left and right with the squeeze action.

Each of the learning sessions for the data presented in this section were run in the following fashion: Leo was first given the slide-box to explore on his own. After approximately 10 minutes, the slide-box is moved to a different location. After approximately 5 more minutes the slide-box is taken away and Leo is presented with the push-box to explore on his own. After approximately 5 minutes, the push-box is moved to a different location and Leo is able to explore it for a final 10 minutes before the experiment ends.

The following is an example of the learning results in the playroom experiment described above. The progression of leo's actions and the creation and generalization of $T \in Tasks$ is depicted in Figure 3-13. Leo is presented with a box, the slide-box. When the system first comes online, the Explore Action is triggered (due to novelty low and activity low) and Leo tries various actions on the box. When he does the flip action, the switch on the box flips from OFF to ON. This state change causes an increase in the novelty drive, and after a few seconds this triggers the Novelty Action and a task is created about this state (T1 in Fig. 3-13). As the state of the world remains constant the novelty drive decreases and after a few seconds exploration continues. Now that the switch is ON, the slide-box is able to open, and when Leo does a sliding motion to the right the lid on the box opens. Leo creates a task about this state change and when it is incorporated into the task set the action policy is extended to include the previous step in the opening task (T2 in Fig. 3-13).

Again once novelty decreases exploration continues and Leo performs various actions with the box in the open state. Doing the flip action again, he makes the switch turn OFF. Later another flip action makes the switch turn ON again. This is a novel state change because the box lid is now open, and it causes a task to be created. When this task is being incorporated into the task set, it meets the criteria for between-policy generalization. Thus, the general task is created (T4 in Fig. 3-13) and the two specific tasks (T1 and T3 in Fig. 3-13) are removed from the task set.

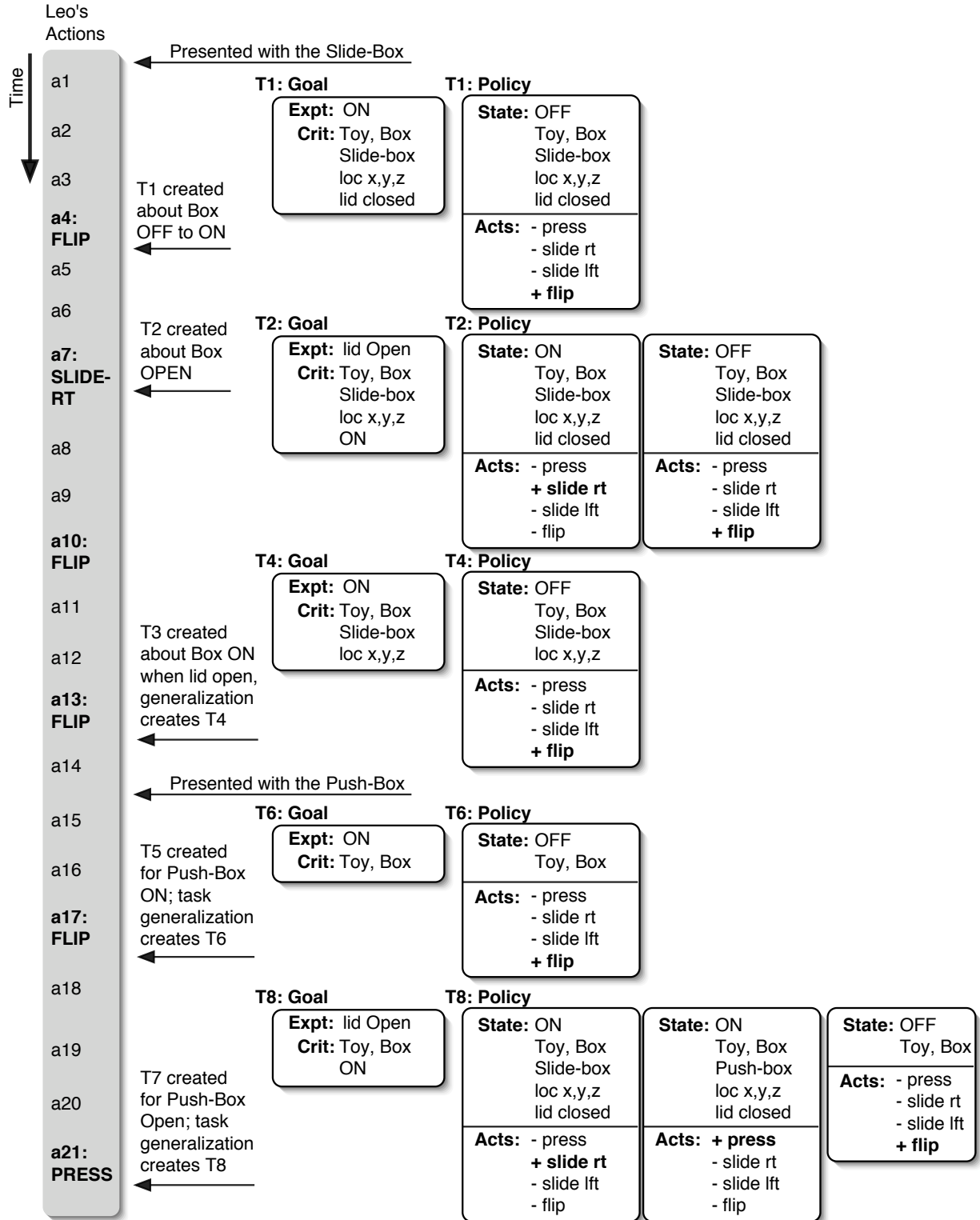


Figure 3-13: Guided Exploration learning example: Leo learns about opening two different kinds of boxes. He is able to generalize about flipping a switch ON (T1, T3, T4, T5, and T6), he learns to open each one (T1, T7) and between-policy generalization makes a general task about opening with the specific policies, within-policy generalization simplifies it further (T8). Due to space, some of the intermediate tasks are not pictured.

After some time, the human partner brings out the push-box toy. Recall, it has a similar switch mechanism, but this toy has a pressing mechanism rather than a sliding mechanism for opening and closing. After some exploration Leo learns to make the switch on this box turn ON and OFF, causing further generalized representations (T6 in Fig. 3-13). And finally, when Leo makes the box lid open with a pressing motion, a task is created for this novel state, and it does meet the criteria for generalization with the previous opening task. However, the action policies for the two tasks are not able to generalize since one uses a sliding motion and the other uses a pressing motion. Thus, the goal is generalized and both specific policies are added to the policy of the new task (T8 in Fig. 3-13).

A human can influence and guide this learning process. They can help define which states are good landmarks, for which a task should be created, by labeling the task (e.g., “Leo, it’s Open!”). They can guide the exploration process by suggesting actions for leo to try (e.g., “Leo, try to Flip the Box”). And throughout the process the human partner can structure the environment and the experience to allow for generalization. Thus, intrinsic measures along with extrinsic support define goals for the machine, and action policies are learned in a standard way for reaching these goals.

The drives are essentially creating a good learning environment for a relatively standard reinforcement learning process. Figure 3.4.8 shows a snapshot of approximately 10 minutes of a learning session. The top graph shows the dynamics of the motivational drives and the bottom graph shows the resulting dynamics of the three learning behaviors. The segment starts with a period where more relevance actions are being triggered, and mastery starts to rise. Then the system is driven to explore, and gets into an area of the world where its mastery is low. This period of exploration is interspersed with learning about novel states, and then more practicing is seen.

The motivational drives create multiple learning opportunities. Additionally the generalization mechanism allows the system to better refine when these tasks can be applied. Figure 3.4.8 shows how the size and content of the set *Tasks* grows and changes over the experimental learning session. The ‘OrigTasks’ series of data shows the number of $T \in Tasks$ that exist in their original form as created by the novelty action (i.e., these are very specific representations, often including a specification of location and other features not relevant to the goal). In the ‘GenTasks’ series we see the number of $T \in Tasks$ over time that are a generalized version (i.e., they are a result of either between-policy or within-policy generalization). Initially, the OrigTasks number increases as new tasks are learned about the slide-box. Over time generalization begins to happen, shown as GenTasks increases and the OrigTasks number decreases. Then halfway through the training session, when the push-box is introduced, a number of new tasks are created

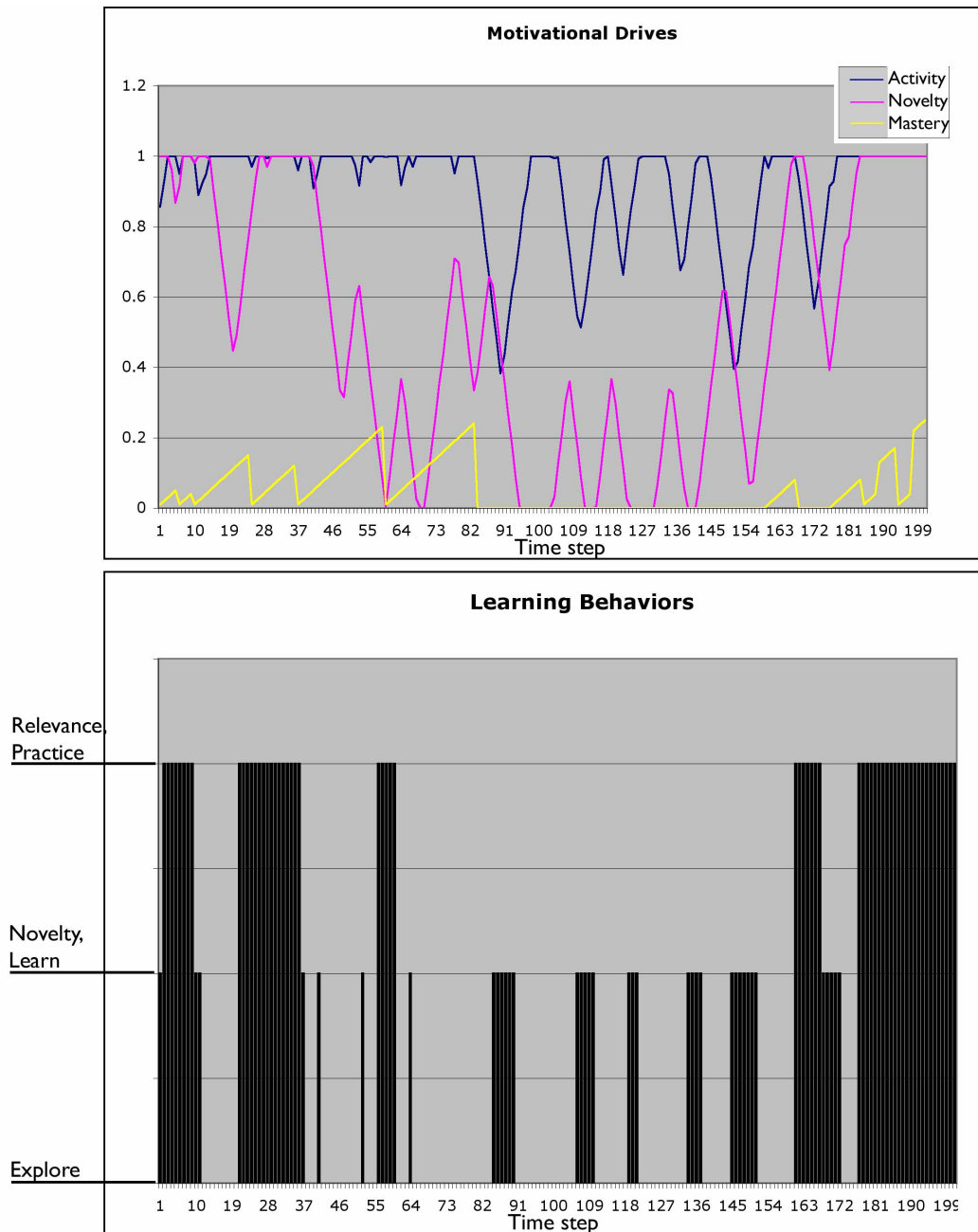


Figure 3-14: A snapshot of approximately 10 minutes of a learning session. The top graph shows the dynamics of the motivational drives and the bottom graph shows the resulting dynamics of the learning behaviors. This segment starts with a period where more Relevance actions are being triggered, and mastery starts to rise. This is followed by a period of exploration interspersed with learning about novel states, and then more practicing is seen.

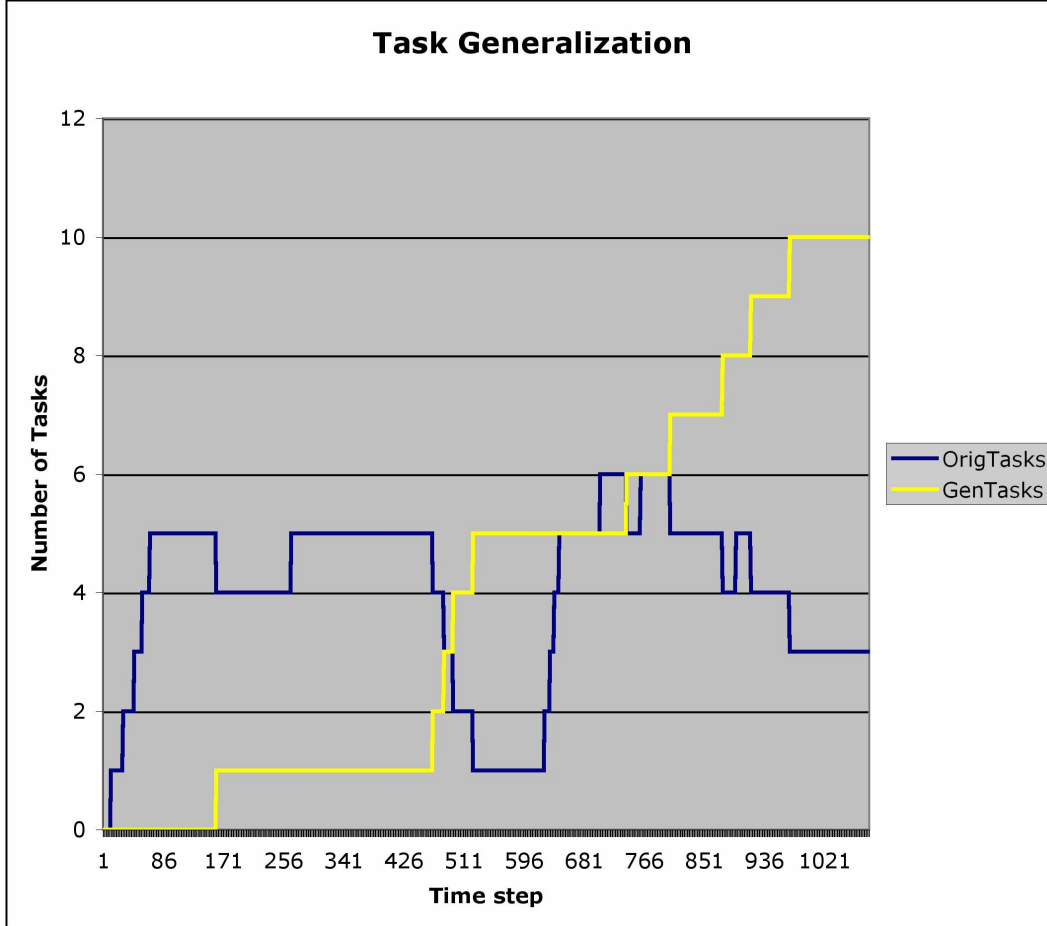


Figure 3-15: An experimental learning session in the virtual playroom. The graph shows how the size of the set $Tasks$ grows and changes over time. In ‘OrigTasks’ series of data shows the number of $T \in Tasks$ that exist in their original form as created by the novelty action. In the ‘GenTasks’ series we see the number of $T \in Tasks$ over time that are a generalized version. Initially, the OrigTasks number increases as new tasks are learned, and as generalization begins to happen, GenTasks increases and OrigTasks number decreases. Then halfway through the training session, when a new object is introduced, a number of new tasks are created so OrigTasks increases again, but then decreases as these also become generalized with experience. After a 25 minute training session, very few $T \in Tasks$ are in their original formulation, they have been refined and generalized through experience and practice.

and OrigTasks increases again. It decreases as these also become generalized with experience. By the end of the 30 minute training session, very few $T \in Tasks$ are in their original formulation, they have been refined and generalized through experience and practice.

3.5 Human Guidance for Machine Learning Systems

Robotic and software agents that operate in human environments will need the ability to learn new skills and tasks ‘on the job’ from everyday people. It is important for designers of learning systems to recognize that while the average consumer is not familiar with machine learning techniques, they are intimately familiar with various forms of social learning (e.g., tutelage, imitation, etc.).

The initial experiment in Chapter 2 with *Sophie’s Kitchen* found people’s desire to guide the character to an object of attention, even when explicitly told that only feedback messages were supported. This raises an important research question for the machine learning community. How do we design machines that learn effectively from human guidance? What is the right level of human interaction at a given time?

It is useful to characterize the level of human interaction as a spectrum from guidance to exploration. On the guidance end of the spectrum is a system that is completely dependent on a human instruction and guidance, and on the exploration end is a system that learns through self exploration with little input from a human partner. In prior works that introduce a human to a machine learning process, the level of human interaction generally remains constant throughout the learning task, remaining at a static point on the guidance-exploration spectrum. This chapter has investigated three points on the guidance-exploration spectrum. Exploring ways in which machines can be designed to more fully take advantage of social guidance in a human teaching interaction.

First, on the guidance end of the spectrum, is Leo’s learning within a Social Dialog. The system builds goal-oriented task representations based on known actions and tasks. It uses social cues that are relevant and understandable to the human partner to frame the learning task. A hypothesis space of goal representations is expanded for a learned task, and through a tightly coupled dialog with a human partner, the best hypothesis is found over a few examples.

Second, on the opposite end of the spectrum, the incorporation of guidance into the interactive Q-Learning agent. In their guidance communication, in the initial experiment with *Sophie’s Kitchen*, people meant to bias the action selection mechanism of the RL algorithm. Introducing a separate interaction channel for attention direction and modifying the action selection mechanism of the algorithm produces a significant im-

provement in the agent’s learning performance. Guidance allows the agent to learn tasks using fewer executed actions over fewer trials. Our modifications also lead to a more efficient exploration strategy that spent more time in relevant states. A learning process, as such, that is seen as less random and more sensible will lead to more understandable and believable agents. Guidance also led to fewer failed trials and less time to the first successful trial. This is a particularly important improvement for interactive agents in that it implies a less frustrating experience, creating a more engaging interaction for the human partner.

Finally, recognizing that both guidance and exploration have their benefits, the Guided Exploration learning with Leonardo brings these together in one learning system. The system has motivations to explore its environment and is able to create goal-oriented task representations of novel events. Additionally this exploration process can be influenced by a human partner in a number of ways: attention direction, action suggestions, labeling of goal states, and positive and negative feedback.

The Guided Exploration version of Leonardo offers many benefits over the Social Dialog version of Leo. The interaction is more flexible, not depending on particular utterances from the human partner. The system is able to learn on its own, and learning is a part of all activity rather than a specific activity triggered by “Leo, let’s learn to X.” Since the human is not marking the start and stop points of a task, the Guided Exploration learner creates tasks for end states and expands the policy back from the goal. Thus the system has to frame its own learning problems.

Many prior works that have a machine learn a new task or skill assume that a goal is known (defined by the designer), is implicit in the reward function given to the learner, or the goal is to learn a complete world model. Alternatively, both the Social Dialog and the Guided Exploration implementations do not make this assumption; instead we ask how a learner can be motivated to learn new tasks/goals with a human partner. A goal-oriented approach to learning is a fundamental capability necessary for social learners, due to the fact that their social partners will act and interpret action in intentional and goal-oriented ways. An SG-ML system will need to continually work to refine the concept of what the human partner has meant to communicate, what the activity is *about*.

Chapter 4

Transparency to Guide a Human Teacher

In a situated learning interaction, the teaching and learning processes are intimately coupled. A good instructor maintains a mental model of the learner's state (e.g., what is understood so far, what remains confusing or unknown, etc.) in order to provide appropriate scaffolding to support the learner's current needs. In particular, attention direction is one of the essential mechanisms that contribute to structuring the learning process [Wertsch et al., 1984]. Other scaffolding acts include providing feedback, structuring successive experiences, regulating the complexity of information, and otherwise guiding the learner's exploration. In general, this is a complex process where the teacher dynamically adjusts their support based on the learner's demonstrated skill level and success.

The learner, in turn, helps the instructor by making their learning process *transparent* to the teacher through communicative acts (such as facial expressions, gestures, gaze, or vocalizations that reveal understanding, confusion, attention), and by demonstrating their current knowledge and mastery of the task [Krauss et al., 1996, Argyle et al., 1973]. Through this reciprocal and tightly coupled interaction, the learner and instructor cooperate to simplify the task for the other — making each a more effective partner.

This chapter investigates several ways in which the transparency of learning and the dynamics of the teacher-learner interaction can positively impact the performance of a machine learning agent. First, the benefit of using gaze to reveal uncertainty is shown with the *Sophie's Kitchen* platform. Then various nonverbal behaviors on Leonardo, used in the implementations described in Chapter 3, are detailed. Finally, a human subject experiment with Leonardo shows that the use of transparency behaviors significantly improves a real-time interactive learning session.

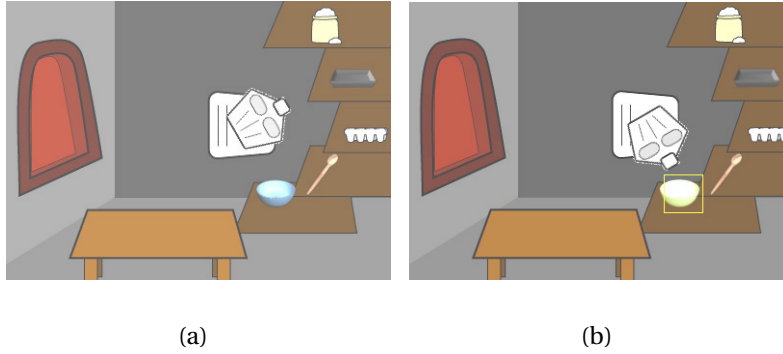


Figure 4-1: Two figures illustrating Sophie's gazing transparency behavior. In Fig. 4-1(a) Sophie is facing the shelf, gazing at the tray prior to selecting a next action; in Fig. 4-1(a) at the bowl.

4.1 Effects of Transparency in *Sophie's Kitchen*

In Chapter 3, we saw that the ability for the human teacher to direct the Sophie agent's attention has significant positive effects on several learning performance metrics (less actions and trials required to complete the task, less failures encountered overall, and a more efficient exploration of the state space). This section reports a related result – that the ability of the agent to use gaze as a transparency behavior results in measurably better human guidance instruction.

4.1.1 Sophie's Gazing Behavior

Gaze requires that the learning agent have a physical/graphical embodiment that can be understood by the human as having a forward heading. In general, gaze precedes an action and communicates something about the action that is going to follow. In this way gaze serves as a transparency device, allowing an onlooker to make inferences about what the agent is likely to do next, their level of confidence and certainty about the environment, and perhaps whether or not guidance is necessary. A gaze behavior was added to the *Sophie's Kitchen* game. The modified game was deployed on the World Wide Web, and data was collected from over 75 people playing the game, allowing for a concrete analysis of the effects Sophie's gaze had on a human teacher's behavior.

Recall the interactive Q-Learning algorithm modified for guidance (Algorithm 2 introduced in Chapter 3). The gaze behavior modification makes one alteration to the stage as which the agent is waiting for guidance, shown in Algorithm 4. When the agent is waiting for guidance, it finds the set of actions, A^* , with the highest Q-values, within a bound β . $\forall a \in A^*$, the learning agent gazes for 1 second at the object-of-attention of a (if it has one). For an example of how the Sophie agent orients towards an ob-

Algorithm 4 Interactive Q-Learning with guidance and a gazing transparency behavior.

```
1: while learning do
2:    $A^* = [a_1 \dots a_n]$ , the  $n$  actions from  $s$  with the highest  $Q$  values within a bound  $\beta$ 
3:   for  $i = 1 \dots n$  do
4:      $o$  = the object of attention of  $a_i$ 
5:     if  $o \neq \text{null}$  then
6:       set gaze of the agent to be  $o$  for 1 sec.
7:     end if
8:   end for
9:   if receive human guidance message then
10:     $g = \text{guide-object}$ 
11:     $a$  = random selection of actions containing  $g$ 
12:   else
13:     $a$  = random selection weighted by  $Q[s, a]$  values
14:   end if
15:   execute  $a$ , and transition to  $s'$ 
    (small delay to allow for human reward)
16:   sense reward,  $r$ 
17:   update policy:
```

$$Q[s, a] \leftarrow Q[s, a] + \alpha(r + \gamma(\max_{a'} Q[s', a']) - Q[s, a])$$

```
18: end while
```

ject to communicate gazing, see Fig. 4-1. This gazing behavior during the pre-action phase communicates a level of uncertainty through the amount of gazing that precedes an action. It introduces an additional delay (proportional to uncertainty) prior to the action selection step, both soliciting and providing the opportunity for guidance messages from the human. This also communicates overall task certainty or confidence as the agent will speed up when every set, A^* , has a single action. The hypothesis is that this transparency will improve the teacher's model of the learner, creating a more understandable interaction for the human and a better learning environment for the agent.

4.1.2 Experimental Design

The *Sophie's Kitchen* game was deployed on the World Wide Web, and participants were solicited to play a computer game, in which their goal was to get the virtual robot to learn how to bake a cake on her own. Participants were told they could not tell Sophie what actions to do, nor could they do any actions directly. They were only able to send Sophie various messages with the mouse to help her learn the task. Depending on their test

Table 4.1: 1-tailed t-test showing the effect of gaze on guidance. Compared to the guidance distribution without gaze, the gaze condition caused a decrease when uncertainty was low and an increase when uncertainty was high. (uncertainty low = number of action choices ≤ 3 , high = number of choices ≥ 3).

Measure	Gaze-Guide	Guidance	t(51)	p
% Guidance when uncertainty low	79	85	-2.22	<.05
% Guidance when uncertainty high	48	36	1.96	<.05

condition, subjects were given instructions on administering feedback and guidance.¹² Each of the participants, played the game once in one of the following conditions:

- **Guidance:** Players were able to use both the feedback and the guidance channels of communication.
- **Gaze-guide:** Players had the feedback and guidance channels. Additionally, the agent used the gaze transparency behavior.

The system maintained an activity log and recorded time step and real time of each of the following: state transitions, actions, human rewards, guidance messages and objects, gaze actions, disasters, and goals. These logs were analyzed to test the following hypothesis:

- **Transparency Hypothesis:** Learners can help shape their learning environment by communicating aspects of the internal process. In particular, the gaze behavior will improve a teacher’s guidance instruction.

4.1.3 Result: Gaze Improves Guidance

This hypothesis is evaluated through the comparison of players that had the guidance condition versus those that had the gaze-guide condition. These results are summarized in Table 4.1. Note that the players that did not have the gaze behavior still had ample opportunity to administer guidance; however, the time that the agent waits is uniform throughout.

Looking at the timing of each player’s guidance instruction, their communication can be separated into two segments: the percentage of guidance that was given when the

¹Full protocol, instructions and consent forms for the study can be found in Appendix A.

²Participation over the web was anonymous and we did not collect gender statistics of the population.

number of action choices was ≥ 3 (high uncertainty), and when choices were ≤ 3 (low uncertainty), note that these are overlapping classes. Three is chosen as the midpoint because the number of action choices available to the agent at any time in the web-based version of *Sophie's Kitchen* is at most 5. Thus we describe a situation where the number of equally valued action choices is ≥ 3 as high uncertainty, and ≤ 3 as low uncertainty.

Players in the gaze-guide condition had a significantly lower percentage of guidance when the agent had low uncertainty compared to the players in the guidance condition, $t(51) = -2.22, p = .015$. And conversely the percentage of guidance when the agent had high uncertainty increased from the guidance to the gaze-guide condition, $t(51) = 1.96, p = .027$. Thus, when the agent uses the gaze behavior to indicate which actions it is considering, the human trainers do a better job matching their instruction to the needs of the agent throughout the training session. They give more guidance when it is needed and less when it is not.

4.2 Nonverbal Transparency Devices on Leonardo

The experiments with the *Sophie's Kitchen* game show that even with an agent that is not designed to be very human-like, people use a social model to make sense of the interaction. The Leonardo platform, on the other hand, was specifically designed for expressive nonverbal communication to participate in natural social interactions with a human partner. The face alone has over 20 actuators (degrees of freedom). For the purpose of Socially Guided Machine Learning, this gives Leo a richer set of behaviors to cooperate in the teaching-learning collaboration. This expressive behavior allows the robot to maintain a mutual belief with the teacher about the task state, expressing confusion, understanding, attention, etc. This section describes the transparency devices Leonardo uses to facilitate the social learning mechanisms described in Chapter 3, and provides an evaluation showing the positive effects such devices have on a learning interaction with human subjects.

4.2.1 Social cues for Scaffolding

A number of expressive skills contribute to Leo's effectiveness in the version of Leonardo that learns in a Social Dialog. Many of these cues are designed around speech act theories and theories of how humans use language to communicate within a joint activity [Clark, 1996]. In particular, principles of grounding. In all activity, humans look for evidence that their action has succeeded, and this extends to joint activity as well. Thus,

Table 4.2: Social Cues for Scaffolding

Context	Leo's Expression	Intention
Human points to object	Looks at Object	Shows Object of Attention
Human present in workspace	Gaze follows human	Shows social engagement
Executing an Action	Looks at Object	Shows Object of Attention
Human: "Let's learn task X"	Subtle Head Nod	Confirms start of task X
Human: "Task X is done"	Subtle Head Nod	Confirms end of task X
Any speech	Perks ears	Conveys that Leo is listening
Speech did not parse	Confusion gesture	Communicates problem
Unconfident task execution	Glances to human more	Conveys uncertainty
Completion of demonstration	Perks ears, lean forward	Soliciting feedback from teacher
Human: "Can you...?"	Perform or Nod/Shake	Communicates task knowledge
Human: "Do task X"	Performs X	Demonstrates representation of X
Task done; Human: "Not quite"	Subtle nod	Confirms, and expects refinement
Task done; Human: "Good!"	Nods head	Confirms task hypothesis
Human asks yes/no question	Nod/Shake	Communicates knowledge/ability
Request is made for an unknown object	Confusion gesture	Communicates problem
Label command has no pointing gesture	Confusion gesture	Communicates problem
Between requested actions	Idle body motion	Creates aliveness
Intermittent	Eye blinks	Creates aliveness
Intermittent	Shifts in gaze	Conveys awareness

the ability to establish joint closure—the mutual belief that a joint activity has succeeded—is fundamental to the success of a collaborative activity. Table 4.2 highlights a number of the social cues that Leonardo uses to facilitate the collaborative activity of learning.

Eye gaze establishes joint attention, reassuring the teacher that the robot is paying attention to the right object at the right time. Subtle nods acknowledge task stages, confirming a mutual understanding of moving on to the next stage when, for instance, the teacher labels a goal state or says a task is complete.

In a realistic robot interaction, the speech recognition system is not perfect and will occasionally not be able to parse the human's utterance. To naturally overcome this roadblock Leo perks his ears as soon as the human begins speaking to indicate that he is paying attention. If unable to parse this speech, Leo will gesture (leaning forward with hand to ear) to indicate that speech recognition failed and the human needs to repeat their last phrase.

The robot uses expressions to indicate to the human tutor when he is ready to learn



Figure 4-2: The extreme poses representing the extent of Leo's emotional facial expression used for transparency in motivated learning with guided exploration.

something new, and demonstration of taught actions provides immediate feedback about task comprehension. When performing a recently taught task, ear and body position as well as eye gaze are used to solicit feedback from the human when uncertainty is high. By frequently looking back at the human during the performance, Leo signals to the teacher that confidence is low, soliciting feedback and further examples.

4.2.2 Facial Expressions to Reveal Internal Learning State

In the Guided Exploration version of Leonardo, there are additional elements of transparency used in the learning process. Emotional expression is used as subtle and natural expression of the state of the learning process. Fig. 4-2 shows the extreme characteristic poses of Leo's facial expression, organized roughly in a two-dimensional space of arousal and valence. The system can blend between these characteristic poses, creating a rich space of facial expression.

Table 4.3: This table is a summary of a table from [Smith and Scott, 1997], showing the various proposed meanings (pleasantness, goal obstacle/discrepancy, anticipated effort, attentional activity, certainty, novelty, personal agency/control) of several individual facial action units. (+) indicates that the facial action is hypothesized to increase with increasing levels of the meaning; (-) indicates that the facial action is hypothesized to increase with decreasing levels of the meaning. These meanings inspire the facial expressions chosen to act as transparency devices in Leo's Guided Exploration.

Facial Action	Proposed Meaning (from Smith and Scott 1997)
Eyebrow frown	-pleasantness, +goal obstacle, +anticipated effort
Raise eyebrows	+attentional activity, +novelty, -certainty, -personal agency/control
Raise upper eyelid	+attentional activity, +novelty, -personal agency/control
Raise lower eyelid	+certainty
Lip corners	+pleasantness
Open mouth	+pleasantness, +attentional activity, -personal agency/control
Tighten mouth	-pleasantness

One approach is to make a calculation of the overall system arousal and valence and have the face continually express these variables. However, in practice, doing so led to a general dulling of emotional expression such that the facial pose remained fairly average all the time. An alternative approach was devised, in which a full characteristic pose is executed but for fleeting moments (2-3 seconds), indicating an internal state and quickly blending back to the neutral pose. The poses are chosen to communicate information to the human partner in a natural way, and this is inspired by research indicating that different facial action units communicate specific meanings [Smith and Scott, 1997] (summarized in Table 4.3). For example, that raised eyebrows and wide eyes indicate heightened attention; and, this is the information we want to communicate with Leo's surprised expression. This approach results in a dynamic, expressive, and informative facial behavior.

Recall the Task Learning Action Group from Chapter 3. There are a number of contexts in which the learning group will trigger action. Leonardo attempts to subtly communicate these trigger contexts to the human partner through facial expression. Table 4.4 lists the learning contexts that trigger fleeting facial expressions. When triggered by a novel event, there is a fleeting surprised expression to let the human know that a task is being formed about this state. When mastery is the trigger, a particular known task is relevant and will be practiced. In this case, Leonardo makes a concentrated facial expression and later makes a happy or sad expression upon the success or failure of this attempt. Throughout the learning process, if the human gives good or bad feedback,

Table 4.4: Leonardo's Facial Expressions to Reveal Learning State in the Guided Exploration implementation.

Context	Facial Expression	Intention
Novel event	Surprised (raised brows/lids and ears, open mouth)	Task being formed about this state.
Mastery triggers execution	Concentration (brows/ears down)	A known task is being tried.
Successful task attempt	Happy (open mouth, raised ears)	Expectation was met
Failed task attempt	Sad (closed mouth, ears down)	Expectation was broken
Good/Bad feedback	Happy/Sad	Acknowledges feedback
Human labels goal state	Happy with head nod	Acknowledges task label

Leonardo makes a happy or sad expression to let the human know they were heard. When the human labels a goal state Leonardo will make a happy expression and also give a head nod to acknowledge the labeling.

4.3 Effects of Leonardo's Nonverbal Communication

The impact of Leo's nonverbal social cues is explored in an experiment where human subjects guide the robot to perform a physical task using speech and gesture. In the task scenario, the human stands across the workspace facing the robot. The robot platform is as described Sec. 3.1. A room-facing stereo-vision system segments the person from the background and locates her face. A downward facing stereo-vision system locates three colored buttons (red, green and blue) in the workspace. It is also used to recognize the human's pointing gestures. A spatial reasoning system is used to determine to which button the human is pointing. The speech understanding system, using Sphinx [Lamere et al., 2003], has a limited grammar to parse incoming phrases. These include simple greetings, labeling the buttons in the workspace, requesting or commanding the robot to press or point to the labeled buttons, and acknowledging that the task is complete.

4.3.1 Experiment

To test the effects of Leo's nonverbal expressions in cooperative interactions with naïve human subjects, each subject was asked to guide the robot through a simple button task where the subjects first taught the robot the names of the buttons, and then had

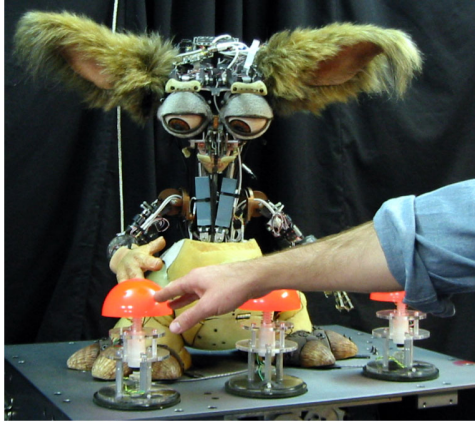


Figure 4-3: Leo and his workspace with three buttons and a human partner.

the robot turn them all on. Although simple, this scenario does provide opportunities for errors to occur: 1) The gesture recognition system occasionally fails to recognize a pointing gesture. 2) The speech understanding system occasionally misclassifies an utterance. Furthermore, errors that occur in the first part of the task (the labeling phase) will cause problems in the second part of the task (the button activation phase) if allowed to go undetected or uncorrected.

Two cases are considered in this experiment. In the transparent case, the robot pro-actively communicates internal states through nonverbal behavior and expressive social cues. In the instrumental case, the robot only does actions instrumental to the task and only communicates internal state when explicitly asked by the human. For instance, in the transparent case, nonverbal cues communicate the robot's attentional state to the buttons and to the human through changes in gaze direction in response to pointing gestures, tracking the human's head, or looking to a particular button before pressing or pointing to it. In addition, the robot conveys liveliness and general awareness through eye blinks, shifts in gaze, and shifts in body posture between specific actions. Its shrugging gestures and questioning facial expression conveys confusion (i.e., when a label command does not co-occur with a pointing gesture, when a request is made for an unknown object, or when speech is unrecognized). Finally, the robot replies with head nods or shakes in response to direct yes/no questions, followed by demonstration if appropriate.

The instrumental case removes the implicit cues that reveal the robot's internal state. Eye gaze does not convey the robot's ongoing attentional focus. Instead, the robot looks straight ahead, but will still look at a specific button preceding a press or point action. There are no behaviors that convey liveliness. The robot does not pro-actively express confusion, and only uses head nods/shakes in response to direct questions.

4.3.2 Procedure

The experiment had 21 subjects from the local campus population (10 males, 11 females), ranging in age from approximately 20 to 40 years. None of the participants had interacted with the Leonardo robot before.

Subjects were first introduced to Leo by the experimenter who pointed out some of the capabilities of the robot and indicated a list of example phrases that the robot understands. These phrases were listed on a series of signs mounted behind the robot. The subject was instructed to complete the following button task with the robot.

1. Teach Leo the names and locations of the buttons.
2. Check to see that the robot knows them.
3. Have Leo turn on all of the buttons. And,
4. Tell Leo that the "all the buttons on task" is done.

Each session was video recorded and the following measures were coded: the total number of errors during the interaction; the time from when an error occurred to being detected by the human; the length of the interaction as measured by time and by the number of utterances required to complete the task. This behavioral analysis tests the following hypotheses:

- H1: The total length of the interaction will be shorter in the transparent case.
- H2: Errors will be more quickly detected in the transparent case.
- H3: The occurrence of errors will be better mitigated in the transparent case.

4.3.3 Results

The analysis offers support for Hypotheses 1 through 3. Of the 21 subjects, video of 3 subjects was discarded. In two of these discarded cases, the robot was malfunctioning to the point where the subjects could not complete the task. In the remaining case, the subject lost track of the task and spent an unusually long time playing with the robot before she resumed the task. Therefore, the video was analyzed for a total of 18 subjects, 9 for the transparent case and 9 for the instrumental case. Table 4.5 summarizes the timing and error results of the video coding.

On average, the total time to complete the button task was shorter in the transparent case, offering support for Hypothesis 1. The average time for the subjects to complete the task in the transparent case is 105 seconds with a standard deviation of 38.0, versus 176 seconds with a standard deviation of 140.9 in the instrumental case. This overall difference is nearly significant ($p = 0.082$).

Table 4.5: Time to complete the overall task as a function of the number of errors (e).

Condition	Category	Errors	Avg Task Time (sec)
transparent	all samples	avg=2.4	105
	$e \leq 1$	max=1	90
	$e > 1$	max=6	112
instrumental	all samples	avg=3.3	176
	$e \leq 1$	max=1	82
	$e > 1$	max=11	293

By breaking each condition into two categories, the low-error trials where one or zero errors occurred and the high-error trials where at least two errors occurred during the interaction, we see that the effect of the transparent case becomes much clearer as the number of errors increases. Analyzing only those trials where at least two errors occurred, the average task time for the transparent case was 112 seconds with a standard deviation of 45.4. In contrast, the average task time for the instrumental case where at least two errors occurred was 293 seconds (over twice as long), with a standard deviation of 138.4. This difference is highly significant ($p = 0.008$).

One reason for the improved overall task time in the transparent condition is the improved robustness during the labeling phase of the task. In the transparent condition, people use the robot’s joint attention ability as an implicit confirmation that the robot learned to associate the correct button with the desired label. Consequently, they can quickly detect a possible labeling error and successfully repair it. Without this visual cue, people spend more time explicitly asking the robot to demonstrate its knowledge of the buttons with “*Can you point to button X*” questions (as shown in Table 4.6). In the transparent condition, subjects generated 1.4 such pointing requests on average, while in the instrumental condition, subjects generated 6.9 requests on average. This difference is significant ($p = 0.015$), supporting Hypothesis 2.

Without the use of gaze as a turn-taking cue, subjects are often much faster in pointing towards and labeling the buttons (at normal adult human speed which is too fast for the robot). Thus in the instrumental condition, provided the gesture recognition system is working well, the time to label all the buttons is quite fast. However, if the gesture system cannot perceive the gesture fast enough or correctly, then the error goes undetected by the human and causes problems in completing the task. As a result, the overall time to label all the buttons is slower in the instrumental condition (see Table 4.6), though this difference is only nearly significant ($p = 0.086$). If we again focus on the trials where at least two errors occurred, the effect becomes much more pronounced:

Table 4.6: Time to complete the labeling portion of the task for each case as a function of the number of errors (e).

Condition	Error	Avg. Point Requests	Avg. Label Time (sec)
transparent	all samples	1.4	57
	$e \leq 1$	0.67	41
	$e > 1$	1.8	65
instrumental	all samples	6.9	125
	$e \leq 1$	4.9	25
	$e > 1$	9.5	249.8

an average labeling time of 65 seconds in the transparent condition versus an average time of 249.8 seconds in the instrumental condition. This difference is highly significant ($p = 0.003$), further support of Hypothesis 2.

Finally, the occurrence of errors appears to be better mitigated in the transparent case, supporting Hypothesis 3. On average, it took less time to complete the task and fewer errors occurred in the transparent case. For the instrumental case, the standard deviation over the number of errors (excluding the error-free trials) is over twice that of the transparent case, showing less ability to mitigate them in the instrumental case. As seen in Table 4.5, more errors occurred in the instrumental case than in the transparent case. Video analysis of behavior suggests that the primary reason for this difference is that the subjects had a much better mental model of the robot in the transparent case due to the nonverbal cues used to communicate the robot's attentional state and when a communication error was likely to occur. The subjects could see when a *potential* error was *about to occur* and they quickly acted to address it.

For instance, in the transparent case, if the subject wanted to label the blue button and saw the robot fix its gaze on the red button and not shift it over to the blue one, the subject would quickly point to and label the red button instead. This made it much more likely for the robot to assign the correct label to each button if the perception system was not immediately responsive. In addition, in the transparent case, the subjects tightly coordinated their pointing gesture with the robot's visual gaze behavior. They would tend to hold their gesture until the robot looked at the desired button, and then would drop the gesture when the robot re-established eye contact with them, signaling that it read the gesture, acquired the label, and was relinquishing its turn.

In summary, when the robot's nonverbal behaviors allowed the human to maintain an accurate mental model of the robot, the quality of teamwork was improved. This transparency allowed the human to better coordinate her activities with those of the robot, either to foster efficiency or to mitigate errors. As a result, the transparent case

demonstrated better task efficiency and robustness to errors. For instance, in viewing the experimental data, the subjects tend to start off making similar mistakes in either condition. In the transparent condition, there is immediate feedback from the robot, which allows the user to quickly modify their behavior, much as people rapidly adapt to one another in interaction. In the instrumental case, however, subjects only receive feedback from the robot when attempting to have it perform an action. If there was an error earlier in the interaction that becomes manifest at this point, it is cognitively more difficult to determine what the error is. In this case, the visual behavior cues in the transparent condition supports rapid error correction in training the robot.

4.4 Transparent Learning Machines

The Socially Guided Machine Learning viewpoint emphasizes the *interactive* elements in teaching. There are inherently two sides to an interaction, and this approach aims to enhance standard machine learning algorithms from both interaction perspectives.

Chapter 3 described several benefits of utilizing social guidance. Recall that, allowing the human teacher to administer guidance in addition to feedback in *Sophie's Kitchen* improves learning performance across a number of dimensions. The agent is able to learn tasks using fewer actions over fewer trials. It has a more efficient exploration strategy that wasted less time in irrelevant states, producing a less random and more sensible exploration which will lead to more understandable and teachable agents. Guidance also led to fewer failed trials and less time to the first successful trial. Additionally social guidance was utilized in various forms with the Leonardo robot. In one implementation the robot participates in a social dialog, allowing a human partner to guide the robot through the completion of a new task and refines its representation over subsequent attempts with the partner. In a second implementation, Leonardo is an exploratory learner and the human partner is able to provide suggestions, feedback, and labels for desired new tasks.

While Chapter 3 dealt mainly with changing the ways that the human is able to interact with the machine learning system, this chapter has detailed the other side of the coin. This chapter has provided concrete examples of how the learning agent can use *transparency* to communicate internal state about the learning process to the human partner. Moreover, when the learning agent does so it improves its learning environment, helping the human partner provide better instruction and guidance.

When the Sophie agent uses gazing behaviors to reveal its uncertainties and potential next actions, people are significantly better at providing more guidance when it is needed and less when it is not. Additionally these transparency behaviors serve to boost

the overall believability of the agent. The issue of believability has been addressed in the animation, video game, and autonomous agent literature for the purpose of creating emotionally engaging characters [Thomas and Johnson, 1981, Bates, 1997]. One contribution of this work is to show how believability relates to teachability of characters to improve the experience of the human and the learning performance of the agent.

The Leonardo platform allows for a richer and more extensive repertoire of social cues. This chapter has described the implementation of several nonverbal behaviors for Leonardo specifically designed to reveal internal state in the Social Dialog and Guided Exploration learning mechanisms. Additionally, significant results of such transparency devices are found in a study with human subjects. When these cues allowed the human to maintain a good mental model of the robot, the quality of teamwork was improved. Transparency allowed the human to better coordinate her activities with those of the robot, either to foster efficiency or to mitigate errors. As a result, the experimental case that utilized transparency devices demonstrated better task efficiency and robustness to errors.

Numerous prior works have explored learning agents (virtual or robotic) that can be interactively trained by people. Many of these works are inspired by animal or human learning. For instance, game characters that the human player can shape through interaction have been successfully incorporated into a few computer games [Evans, 2002, Stanley et al., 2005, Stern et al., 1998]. Animal training techniques have been explored in several robotic agents [Kaplan et al., 2002, Saksida et al., 1998, Steels and Kaplan, 2001]. As a software agent example, Blumberg's virtual dog character can be taught via clicker training, and behavior can be shaped by a human teacher [Blumberg et al., 2002].

Many of these prior works agree with our situated learning paradigm for machines, and have emphasized that an artificial agent should use social techniques to create a better interface for a human partner. This work goes beyond gleaning inspiration from natural forms of social learning and teaching to formalize this inspiration and empirically ground it in observed human teaching behavior through extensive user studies. Thus, another contribution of this work is empirical evidence that social guidance and transparency create a good interface for a human partner, *and* can create a better learning environment that significantly benefits learning performance.

Finally, the scenario of human input has received attention in the machine learning community. There has been work on computational models of teacher-learner pairs [Goldman and Mathias, 1996]. Active learning and algorithms that learn with queries begin to address interactive aspects of a teacher-learner pair [Cohn et al., 1995]. Queries can be viewed as a type of transparency into the learning process, but in these approaches this does not steer subsequent input from a teacher. Instead, through its queries,

the algorithm is in control of the interaction. Cohn *et al.* present a semi-supervised clustering algorithm that utilizes a human teaching interaction, but the balance of control falls to the human (i.e., to iteratively provide feedback and examples to a clustering algorithm which presents revised clusters) [Cohn et al., 2003].

Thus, prior works have addressed how human input can theoretically impact a learning algorithm. In contrast, this work addresses the nature of *real* people as teachers; the ground truth evaluation is the performance of the machine learner with non-expert human teachers. Whereas prior works typically lend control either to the machine or the human, the contribution of this work is the focus on how a machine learner can use transparency behaviors to steer the instruction it receives from a human, creating more reciprocal control of the interaction.

Chapter 5

The Asymmetry of Human Feedback

In the initial experiments with *Sophie's Kitchen*, one of the main findings concerned the biased nature of positive and negative feedback from a human partner (Section 2.3.3). Clearly, people have asymmetric intentions they are communicating with their positive and negative feedback messages.

This chapter addresses the asymmetric meaning of positive and negative feedback. The intuition is that positive feedback tells a learner undeniably, “what you did was good.” However, negative feedback has multiple meanings: 1) that the last action was bad, and 2) that the current state is bad and future actions should correct that. Thus, negative feedback is about both the past and about future intentions for action.

The two implementations in this chapter present two interpretations of negative feedback. Both assume that negative feedback from a human partner is feedback about the action or task performed and at the same time communicates something about what should follow. In the first example, Leonardo assumes that negative feedback will lead to refinement of the performed task example. In the second example, Sophie assumes that a negatively reinforced action should be reversed if possible. This UNDO interpretation of negative feedback shows significant improvements in several metrics of learning performance.

5.1 Negative Feedback Leading to Refinement

Chapter 3 described an implementation that allows the Leonardo robot to learn new tasks within a social dialog. One particular aspect of that implementation, just-in-time error correction, utilizes an asymmetric meaning of positive and negative feedback from a human partner. During the learning dialog, when Leonardo demonstrates a learned task, positive feedback reinforces a task hypothesis, but negative feedback leads directly to refinement of the hypothesis.

This approach is drawn from speech act theory, in particular the concept that speakers intend their larger purposes to be inferred from their utterances [Clark, 1996]. In the case of Leonardo, by gesturing in a way to solicit feedback after a demonstration the robot is asking: “Was that the right thing to do?” It is assumed that if the human answers this question they will infer the larger purpose of the joint activity, which implies some commitment to a more than a yes/no response. If the human were to simply answer “no,” this does not represent a commitment to the larger joint activity of helping Leo correctly learn the task.

5.1.1 Task Execution and Refinement

Recall from Chapter 3, when Leo is asked to do a known task, and the goal is incomplete, Leo uses the current best task hypothesis for execution, which has a likelihood (between 0 and 1) relative to the other hypotheses available. If this confidence is low ($< .5$), Leo expresses tentativeness (frequently looking between the instructor and an action’s object of attention). Upon finishing the task, Leo leans forward with his ears perked waiting for feedback. The teacher can give positive verbal feedback (e.g., “Good,” “Good job,” “Well done,” ...) and Leo considers the task complete and the executed hypothesis gains value (i.e., the number of seen examples consistent with this hypothesis is incremented; thus, $P(D|h)$ increases for this hypothesis in the Bayesian likelihood calculation).

After completing the demonstration, if Leo has not yet achieved the goal the human can give negative verbal feedback (e.g., “No,” “Not quite,” ...) and Leo will expect the teacher to lead him through the completion of the task. A new example is created through this refinement stage, as described in Section 3.2.2. Leo makes a representation of the change over the task and the actions that were necessary to complete it (the actions he did himself, plus the actions the human requested during refinement). Then a space of hypotheses consistent with this refined example is expanded, as described in Section 3.2.3. For each hypothesis, if it already exists in the task hypothesis space then the number of seen consistent examples is incremented, otherwise it is added to the space. Again, with the Bayesian likelihood method, the best hypothesis is chosen for the next execution of this task.

5.1.2 Just-in-Time Correction

The turn-taking dialog framework lets the teacher know right away what problems or issues remain unclear, enabling just-in-time error correction with refinement to failed attempts. Through gesture and eye gaze, the robot lets the teacher know when the current task representation has a low confidence, soliciting feedback and further examples.

Algorithm 5 Interactive Q-Learning with the addition of the UNDO behavior

```
1: while learning do
2:   if (reward last cycle < −.25) and (can undo last action,  $a_{last}$ ) then
3:      $a = \text{undo}(a_{last})$ 
4:   else
5:      $a = \text{random select weighted by } Q[s, a] \text{ values}$ 
6:   end if
7:   execute  $a$ , and transition to  $s'$ 
   (small delay to allow for human reward)
8:   sense reward,  $r$ 
9:   update policy:
```

$$Q[s, a] \leftarrow Q[s, a] + \alpha(r + \gamma(\max_{a'} Q[s', a']) - Q[s, a])$$

```
10: end while
```

A similar goal concept learning could be achieved with a supervised learning approach that uses batches of positive and negative examples to learn the concept. However, this does not take advantage of the tightly coupled interactive component of learning from a human teacher. Leonardo's on-line interactive learning session lets the human partner provide examples incrementally. They see through demonstration the current state of Leo's goal concept, and are able to interactively make additions to a negative example to change it into a positive example of the goal concept.

5.2 Negative Feedback Leading to Action Reversal

The *Sophie's Kitchen* platform is used to explore another aspect of reward asymmetry. In this approach, negative feedback communicates information both to the learning mechanism updating the policy (in the same way as positive rewards), and also to the action selection mechanism. This implementation shows significant improvements in multiple aspects of learning performance with a human partner, allowing the agent to have a more efficient and robust exploration strategy.

Positive reward for an action just performed gives a clear message to the agent - that the probability of performing that action in that state should be increased. A symmetric approach would have the opposite reaction to a negative reward - the probability of performing that action in that state should be decreased. While learning will occur in the symmetric case (the success of several renditions of Reinforcement Learning algorithms are proof), this neglects part of the information communicated by a negative reward.

In addition to communicating that the decision to make that action was wrong, neg-

ative feedback communicates that this line of behavior or reasoning is bad. Thus a reaction that more closely resembles intuition about natural learning, is to adopt the goal of being back in the state that one was in before the negative feedback occurred. In many cases, of course not all, actions performed by an agent in the world are reversible. Thus upon negative feedback that agent should first update its value function to incorporate this feedback from the world, but this negative feedback should also communicate with the action selection mechanism that the next action should be a reversal if possible.

Experiments with the Sophie platform show this behavior to lead to more robust learning, keeping the agent in the positive areas of the world, approaching the boundaries but avoiding the negative spaces. This is particularly important for applications in robotic agents acting in the real world with physical hardware that may not withstand much negative interaction with the world. This behavior also generates more efficient learning, reducing both the total time necessary and the number of trials that end in failure.

5.2.1 Modification for Sophie's UNDO Response

The experiment presented below uses a modification to the interactive Q-Learning algorithm, Algorithm 1. This baseline algorithm is modified to respond to negative feedback with an UNDO behavior (a natural correlate or opposite action) when possible. Thus a negative reward affects the policy in the normal fashion, but also alters the subsequent action selection if possible. The proper UNDO behavior is represented within each primitive action and is accessed with an *undo* function:

- The action GO [direction] returns GO [-direction]
- The action PICK-UP [object] returns PUT-DOWN [object]
- The action PUT-DOWN [object] returns PICK-UP [object]
- The USE actions are not reversible.

Algorithm 5 shows how this is implemented with the changes in lines 2 – 6, as compared to the baseline Algorithm 1.

5.2.2 Evaluation

Experimental data was collected from 97 non-expert human participants by deploying the *Sophie's Kitchen* game on the World Wide Web. They were asked to help the agent learn to bake the cake by sending feedback messages as she makes attempts. When they felt Sophie could bake the cake herself they pressed a button to test the agent and

obtained their score (based on how many actions it took for the agent to bake the cake on her own).^{1 2}

The *Sophie's Kitchen* platform offers a measurable comparison between two conditions of the learning algorithm. In the baseline case the algorithm handles both positive and negative feedback in a standard way, feedback is incorporated into the value function (Alg. 1). In the undo case the algorithm uses feedback to update the value function but then also uses negative feedback in the action selection stage as an indication that the best action to perform next is the reverse of the negatively reinforced action (Alg. 5). Statistically significant differences were found between the baseline and undo conditions on a number of learning performance metrics (summarized in table 5.1).

Training Failure Reduction

The UNDO behavior helps the agent avoid failure. The total number of failures during the learning phase was significantly less in the undo case, $t(96) = -3.77$, $p < .001$. This is particularly interesting for robotic agents that need to learn in the real world. For these agents, learning from failure may not be a viable option; thus, utilizing a negative feedback signal to learn the task while avoiding disaster states is necessary.

The undo case also had significantly less failures before the first goal was reached, $t(96) = -3.70$, $p < .001$. Related to the overall number of failures being less, there were also less failures before the first success. This is especially important when the agent is learning with a human partner. The human partner will have a limited patience and will need to see progress quickly in order to remain engaged in the task. Thus, the undo behavior seems to be a good technique for reaching the first success faster.

Training Time Efficiency

There was a nearly significant effect for the number of actions required to learn the task, $t(96) = -1.32$, $p = .09$, with the undo condition requiring less steps (the high degree of variance in the number of steps needed to learn the task leads to the higher p value). Thus, the algorithm that uses the undo behavior is able to learn the task in less time (fewer total actions taken).

Exploration Efficiency

Another indication of the efficiency of the undo case compared to the baseline is in the state space needed to learn the task. The number of unique states visited is significantly

¹Full protocol, instructions and consent forms for the study can be found in Appendix A.

²Participation over the web was anonymous and we did not collect gender statistics of the population.

Table 5.1: 1-tailed t-test: Significant differences were found between the baseline and undo conditions, in training sessions with nearly 100 non-expert human subjects playing the *Sophie's Kitchen* game online.

Measure	Mean baseline	Mean undo	chg	t(96)	p
# states	48.3	42	13%	-2.26	=.01
# F	6.94	4.37	37%	-3.76	<.001
# F before G	6.4	3.87	40%	-3.7	<.001
# actions to G	208.86	164.93	21%	-2.25	=.01
# actions	255.68	224.2	12%	-1.32	=.095

less in the undo case, $t(96) = -2.26$, $p = .01$. This indicates that when the algorithm interprets negative feedback as a directive for reversing the previous action, or returning to the previous state, the resulting behavior is more efficient in its use of the state space to learn the desired task. Thus, the learning agent stays ‘on the right track’ in its exploration.

5.3 Asymmetric use of Feedback in Machine Learning

In Reinforcement Learning it is usual to represent the distinction between appetitive and aversive evaluative feedback using just the sign of a scalar reward signal, where positive means good; negative means bad. Since RL algorithms are based on the objective of maximizing the sum of rewards over time, this makes sense: positive feedback increases the sum; negative feedback decreases it. But we see from Chapter 2 that when a human partner is asked to train an RL agent, they do not use the reward channel in symmetric ways.

Furthermore, it is clear that biological systems do not have symmetric responses to positive and negative feedback. Evidence from neuroscience shows that the human brain processes appetitive and aversive rewards differently. Positive and negative feedback stimulate physically different locations in the brain: the left side of the amygdala responds to positive reinforcement, while the right responds to negative reinforcement [Zalla et al., 2000]. Additionally, there is evidence for an ‘error processing’ mechanism where the anterior cingulate cortex generates signals correlated with error detection (independent of task goal or modality) [Holroyd and Coles, 2002]. This evidence alone does not tell us how or why to include the asymmetry of feedback in our computational learning model, but it does inspire us to search for computational grounds for such inclusion with the goal of developing more efficient and robust learning algorithms. This chap-

ter has presented two such computational implementations for treating appetitive and aversive feedback differently.

In the first example, the Leonardo robot assumes that a task demonstration followed by negative feedback will lead to refinement of that example. This is a departure from the normal formulation of supervised learning, where the agent receives a bag of positive and negative examples (or perhaps collects these online over time). In this case the agent has seen only positive examples, and expands hypothesis goal representations. Upon executing a task based on one of these hypotheses, and getting negative feedback, Leo expects the human partner to lead him through refining the example. This lets the agent at once label the hypothesis as bad and at the same time add another positive example to its set. Thus refining the hypothesis space with the human partner.

In *Sophie's Kitchen* on the other hand, the agent takes a different view of negative feedback. It assumes that negative feedback should lead to reversing an action if possible. In the kitchen world, many of the actions are reversible, such that the previous state can be easily achieved. If negative reinforcement is received and the last action performed is reversible the agent chooses this as the next action rather than using its normal action selection mechanism. In experiments with human trainers, this version of the Sophie agent shows significantly better learning performance. The size of the state space visited is much less, there are significantly fewer failures, and fewer actions are needed to learn the task.

Finally it is interesting to address the simultaneous use of the two implementations shown in this chapter. At first glance they may seem incompatible, however, the approaches represent two strategies on opposite ends of the guidance-exploration spectrum. In the Leo example the assumption is that *more* needs to be done from the current state and the human partner is guiding the additional steps. On the other hand, the Sophie example shows the utility of reverting to the previous state and trying again. Waiting for refinement is a guidance-oriented response to negative feedback, while 'undo' or 'do over' is an exploration-oriented response to negative feedback. In the end, a learning agent is likely to need the ability to use both strategies, having the ability to slide dynamically along the guidance-exploration spectrum. As seen throughout this thesis, the ideal SG-ML system should be able to both learn on its own but take full advantage of the human partner if they are present and offering support.

Chapter 6

Contributions

This thesis concerns *Socially Guided Machine Learning*, exploring the ways in which machine learning can exploit social learning. The cornerstone of this research is the belief that machines designed to interact with people to learn new things should more fully be able to participate in the teaching and learning partnership, a two-way collaboration. Moreover, the ability to utilize and leverage social interaction is more than a good interface for people, it can positively impact the underlying learning mechanisms to let the system succeed in a real-time interactive learning session.

Typical machine learning techniques have not been specifically designed for learning from untrained users, thus the learning process for standard ML techniques is not currently feasible for non-experts. In Socially Guided Machine Learning, the goal is to understand how to bridge this gap, enabling machine learning systems to succeed at learning within a social interaction with everyday people. This chapter details the specific contributions made in this thesis towards the understanding of Socially Guided Machine Learning.

- An experiment investigating human teaching behavior yields three general characteristics exhibited across participants.
- The guidance-exploration spectrum is a novel characterization of human interaction with machine learning. Three implementations represent several points along this spectrum.
- An implementation and experiment in *Sophie's Kitchen* shows that everyday human trainers are able to use guidance with a Reinforcement Learning agent, resulting in significant performance improvements.
- Implementations of transparency devices to reveal aspects of the internal learning state have been shown with software and robotic agents. Experiments with both

Sophie and Leonardo show that transparency leads to significant improvements in the quality of instruction received from a human teacher.

- Implementations with Sophie and Leonardo represent two asymmetric interpretations of feedback from a human teacher. An experiment with human trainers shows significant positive benefits to the learning mechanism.
- Novel approaches and implementations of goal-oriented task learning have been demonstrated on the Leonardo robot.

6.1 Experimental findings about how people want to teach

This thesis contributes to the understanding of how people approach the task of teaching a machine learner. Numerous prior works have explored learning agents (virtual or robotic) that can be interactively trained by people, reviewed in Chapter 1. Many of these works are inspired by animal or human learning (e.g., game characters that the human player can shape through interaction [Evans, 2002, Stanley et al., 2005, Stern et al., 1998], and animal training techniques for robotic and software agents [Kaplan et al., 2002, Saksida et al., 1998, Steels and Kaplan, 2001, Blumberg et al., 2002]). Many of these prior works are also inspired by a situated learning paradigm for machines, and have emphasized that an artificial agent should use social techniques to create a better interface for a human partner. The work presented in the thesis goes beyond gleaning inspiration from natural forms of social learning and teaching to formalize this inspiration and empirically ground it in observed human teaching behavior through extensive user studies.

The *Sophie's Kitchen* experiment presented in Chapter 2 investigates “how people want to teach” and yields three general characteristics that people exhibited:

- People want the ability to direct the agent’s attention, guiding the exploration.
- Players try to maximize their impact on the learning process as they infer a mental model of the learner.
- Positive and negative feedback from a human teacher have asymmetric intentions or meanings.

6.2 The Guidance-Exploration Spectrum

Chapter 3 introduced a novel characterization of human interaction with machine learning systems, the spectrum of guidance and exploration. As seen in prior works (Sec.

1.3.1) most systems that incorporate a human teacher into the learning process maintain a constant level of involvement of the human partner. Several are highly dependent on the human teacher's guidance, and will learn nothing without their interaction. Others are almost entirely exploration based, and barely take advantage of the human partner. This thesis has addressed the important research question for SG-ML: how to seamlessly incorporate both guidance and exploration, resulting in a system that can learn on its own, but also take full advantage of a human partner if they are there to provide guidance.

Three systems were implemented that explore different points along the spectrum of guidance and exploration. On the guidance end of the spectrum, 'Learning within a Social Dialog' on the Leonardo robot has many desirable SG-ML qualities that allow it to take advantage of natural human guidance within a tutorial dialog. On the exploration end of the spectrum, the *Sophie's Kitchen* game was modified to incorporate human guidance, and an experiment with human subjects quantified the effects of human guidance on a standard exploratory learner. Finally, the lessons from these two systems result in a third learning mechanism, 'Guided Exploration', implemented on the Leonardo robot, in which the learning system uses both guidance and exploration.

6.3 Guidance with Everyday Human Trainers

Prior works have pointed out how supervision or guidance might benefit a machine learner [Clouse and Utgoff, 1992, Smart and Kaelbling, 2002], but in the *Sophie's Kitchen* experiments presented in Chapter 3 we are able to show that ordinary people, given only a high level description of the task and the agent, can understand and utilize a guidance channel to improve the learning performance.

Guidance allows the agent to learn tasks using fewer executed actions over fewer trials. Our modifications also led to a more efficient exploration strategy that spent more time in relevant states. A learning process, as such, that is seen as less random and more sensible will lead to more understandable and believable agents. Guidance also led to fewer failed trials and less time to the first successful trial. This is a particularly important improvement for interactive agents in that it implies a less frustrating experience, creating a more engaging interaction for the human partner.

6.4 Transparency to Improve the Learning Environment

In human learning, teachers direct a learner's attention, structure experiences, support attempts, and regulate complexity. The learner contributes by revealing their internal

state to help guide the teaching process. Each simplifies the task for each other. The findings in the study presented in Chapter 2 support this notion of *partnership*. When everyday users are asked to train a machine learning agent, they adjust their training behavior as the interaction proceeds, reacting to the behavior of the learner.

Chapter 4 provided concrete examples of how the learning agent can use *transparency* to communicate internal state about the learning process to the human partner. Moreover, experiments show that doing so improves its learning environment, helping the human partner provide better instruction and guidance.

When the Sophie agent uses gazing behaviors to reveal its uncertainties and potential next actions, people were significantly better at providing more guidance when it was needed and less when it was not. The Leonardo platform allows for a richer and more extensive repertoire of social cues, detailed in Chapter 4. A study with human subjects shows the significant benefit of these transparency devices. When these cues allowed the human to maintain a good mental model of the robot, the quality of teamwork was improved. Transparency allowed the human to better coordinate her activities with those of the robot, either to foster efficiency or to mitigate errors. As a result, the experimental case that utilized transparency devices demonstrated better task efficiency and robustness to errors.

6.5 Asymmetric Interpretations of Human Feedback

One of the findings of the experiment in Chapter 2 concerned the biased nature of positive and negative feedback from a human partner. The majority of participants gave significantly more positive rewards than negative rewards. Clearly, people have asymmetric intentions they are communicating with the positive and negative feedback channels.

Chapter 5 addressed the asymmetric meaning of positive and negative feedback. The two implementations in this chapter assumed that negative feedback from a human partner is both feedback about the action or task performed and at the same time communicates something about what should follow. In the first example, Leonardo assumes that negative feedback will lead to refinement of the performed task example. In the second example, Sophie assumes that a negatively reinforced action should be reversed if possible. This UNDO interpretation of negative feedback shows significant improvements in several metrics of learning performance. In experiments with human trainers, this version of the Sophie agent shows significantly better learning performance. The size of the state space visited is much less, there are significantly fewer failures, and fewer actions are needed to learn the task.

The approaches represent two strategies on opposite ends of the guidance-exploration spectrum. In the Leo example the assumption is that *more* needs to be done from the current state and the human partner is leading the additional steps. The Sophie example shows the utility of reverting to the previous state and trying again. Waiting for refinement is a guidance-oriented response to negative feedback, while ‘undo’ is an exploration-oriented response to negative feedback. In the end, a learning agent is likely to need the ability to use both strategies, having the ability to slide dynamically along the guidance-exploration spectrum. Again this addresses a fundamental SG-ML goal, that the ideal system should be able to both learn on its own but take full advantage of the human partner if they are present and offering support.

6.6 Mechanisms of Goal-oriented Learning

The implementations in Chapter 3 address several important aspects of goal-oriented learning. In most machine learning examples, learning is an explicit activity. The system is designed to learn a particular thing at a particular time. With human learning, on the other hand, there is a motivation for learning, a drive to improve, and an ability to seek out the expertise of others.

Thus, as a departure from a standard machine learning approach, the Guided Exploration implementation described in Chapter 3 has motivations for learning that underlie all activity: novelty, mastery and activity drives. These competing drives create an exploration behavior that creates learning opportunities for the agent to learn on its own, but also drive the motivation to take advantage of a human partner when they are available.

Additionally, in most machine learning examples, in particular examples that have a system learn a new task or skill, it is often assumed that the system is given the task goal or criteria function. This work backs off of that assumption and addresses how a learner can be motivated to learn new tasks/goals online with a human partner.

The motivational drives create a good learning environment for a relatively standard reinforcement learning process. An options learning mechanism is augmented with a generalization mechanism that allows the system to better refine when a learned task can be applied. The human scaffolding lets the system define landmarks and goals along the way rather than the designer having had to encode this into the reward function, and the human partner structures the environment and the experience to allow for appropriate generalization. Thus, intrinsic measures along with extrinsic support define goals for the machine, and action policies are learned for reaching these goals. This goal-oriented approach of having a reinforcement learner define what options are good to know, framing its own learning problems, is novel and is fundamental for a social learner.

6.7 Concluding Remarks

In Socially Guided Machine Learning, we advocate designing for the performance of the complete, coupled human-machine teaching-learning system. This thesis has made several contributions towards the understanding of Socially Guided Machine Learning, covering several fundamental SG-ML topics. This new perspective reframes the machine learning problem as an interaction between the human and the machine, and allows us to take advantage of human teaching behavior to construct a machine learning process that is more amenable to the human partner. This interaction approach to machine learning forces the research community to consider many new questions. Some of the grand challenges ahead for SG-ML include:

- Goal-oriented exploration, exploitation, and experimentation: In order for a system to be guidable by an everyday person, the exploration process must be understandable. This thesis has shown several ways to achieve a more understandable exploration, and in future work, this line of research can be taken further. For example, imagine an experimentation extension to the Explore Action of the Guided Exploration on Leonardo. Rather than posing the problem as a tradeoff between exploring and exploiting, for a goal-oriented learner perhaps we need also to include experimenting. Thus, the system would be able to explore completely new territory, exploit and practice known tasks and skills, and falling between these two, the system could alter its known tasks slightly to experiment with their boundaries and applicability in new domains.
- Mixed-initiative learning: In the learning examples of this thesis the machine learns a new task through its own experience (guided or instructed by the human partner at times). An important line of future work involves combining the merits of learning by observation techniques with the kinds of learning through experience techniques contributed here. In a SG-ML scenario, the machine will likely need the ability to participate in a mixed-initiative learning interaction, fluidly switching between watching and acting, in order to learn a new task.
- Appraisal mechanisms: Several areas of future work exist in the study of ways that an SG-ML system should accurately appraise its environment and its behavior. Incorporating an emotion system into the cognitive architecture would be a cognitively realistic approach to appraising the internal and external environment. Additionally, this would allow for the use of affective regulation of the learning process. As one example this could influence the probability of giving up in the Relevance Action, or breadth versus depth in an exploration process. Another area

of study involves how a machine might learn intrinsic measures of success.

- Mechanisms of engagement: In order to remain engaged over long periods of time, the teaching process has to be rewarding for the human. This thesis has shown a few ways that the learning process can be made more engaging for the human partner, but a fruitful area of future work is in exploring various mechanisms of engagement for the learning process. Visible progress and social connection are two elements that might strengthen engagement in the machine learning process.

This research agenda will enable a number of exciting future applications. For example, personal robot assistants in everyday human environments will require SG-ML capabilities. When robots are able to learn via social interaction from ordinary people this will enable them to be usefully deployed in everyday human environments. People in their homes, schools, hospitals and offices will be able to teach these robots to perform new tasks to help them achieve their goals. For instance, a robot that can help an elderly person remain self-sufficient in their own home, or a robot that can be a cooperative partner in a home improvement project. It would be impossible for a designer to encode into the machine ahead of time every skill necessary to achieve these types of goals. A machine that learns opportunistically through self-motivated exploration will also offer a new kind of educational technology. Such a robot could be a true learning companion for a child, creating a co-learning scenario where the robot and the child are exploring the environment together, learning from each other's discoveries. It's also important to recognize that teaching is a fundamentally rewarding activity for us as humans, thus teachable machines and software agents will usher a new realm of entertainment technology. SG-ML technology will enable teachable characters for a novel genre of computer and robotic games.

In aiming to enable robots and machines in general to learn new tasks from natural human instruction with ordinary people (not experts in robotics or machine learning), it will be important to enable these systems to take advantage of social interactions. Structuring guidance through interpersonal interaction will be natural for everyday people who need to teach their machines new things. We need a principled theory of the content and dynamics of this tightly coupled teaching-learning process in order to design systems that can learn efficiently and effectively from ordinary users. This thesis has made several contributions towards the understanding of Socially Guided Machine Learning, explicating the fundamental SG-ML principles of Guidance, Transparency, Asymmetry, and Goal-Oriented Learning.

Appendix A

Sophie's Kitchen Experiments

Throughout this thesis various experiments were completed with the Sophie's Kitchen platform. This appendix will cover the details of exactly how these experiments were run, both in lab and online, including system configuration difference, instructions, payments, and consent forms.

A.1 Experiment 1 – in Lab

In the initial experiments in lab, covered in Chapter 2, participants were solicited from the campus community via email and completed the experiment in the lab space of the Robotic Life Group and the MIT Media Lab (E15-468).

A.1.1 Experimental Protocol

- Introduction: Participants will be given a short introduction to the study and given the informed consent form.
- Game Task: Participants will be asked to play a video game. It is expected this will last for approximately 20-40 minutes (though the time spent is entirely up to the participant and is one of our measures). In the video game there is a virtual robot character that is in a scene with a number of everyday objects. After being shown the game, participants will be given a task that they are to get the robot to learn how to do (one example: in a kitchen scene they may be asked to teach the robot the proper sequence of steps involved in baking a cake given objects like bowls, spoons, sugar, flour, an oven, etc.). The robot character has 'a mind of its own' and when told to begin it will try to start guessing how to do the task. The participants will have to communicate with the character to let it know when it is doing good or bad until it has the right idea for how to complete the goal task.

- Questionnaire: Once the participant is done they will complete a questionnaire.
- Payment: At the end the participant will receive payment. \$5 for participation, and an additional amount (up to \$10) based on the performance of their character on the goal task, based on a demonstration completed after they indicate they are finished teaching.

A.1.2 Informed consent signed by each participant

You are asked to participate in a research study conducted by Cynthia Breazeal (Associate Professor), Guy Hoffman (Ph.D. candidate) and Andrea Thomaz (Ph.D. candidate), from the Robotic Life Group at the Massachusetts Institute of Technology (M.I.T.). Results of this study will contribute to the Ph.D. thesis research of Guy Hoffman and Andrea Thomaz. You should read the information below, and ask questions about anything you do not understand, before deciding whether or not to participate.

PARTICIPATION AND WITHDRAWAL

Your participation in this study is completely voluntary and you are free to choose whether to be in it or not. If you choose to be in this study, you may subsequently withdraw from it at any time without penalty or consequences of any kind. The investigator may withdraw you from this research if circumstances arise which warrant doing so.

PURPOSE OF THE STUDY

We are investigating Machine Learning applications for software computer games.

PROCEDURES

If you volunteer to participate in this study, we would ask you to do the following things:

You will be asked to play a video game, in which your goal is to train the virtual robot character to complete one of a variety of tasks. You will be able to communicate with the character through the use of the keyboard and the mouse. Once you feel your character has learned the task, you will complete a questionnaire about the experience. The complete study is estimated to take less than one hour of your time.

POTENTIAL RISKS AND DISCOMFORTS

We are unaware of any potential risks in this experiment.

POTENTIAL BENEFITS

Your participation will help us to build software agents and robots that are more responsive and sociable learning partners.

PAYMENT FOR PARTICIPATION

Every participant will receive, \$5 for doing the experiment. You can receive up to an additional \$10 based on the speed and accuracy with which your character learns the task.

CONFIDENTIALITY

Any information that is obtained in connection with this study and that can be identified with you will remain confidential and will be disclosed only with your permission or as required by law.

IDENTIFICATION OF INVESTIGATORS

If you have any questions or concerns about the research, please feel free to contact:

Associate Professor, Cynthia Breazeal; 617 452 5601; MIT Media Lab, E15-468, Cambridge, MA 02139; cynthiab@media.mit.edu

Andrea L. Thomaz (Ph.D. candidate); 617 452 5612; MIT Media Lab, E15-48, Cambridge, MA 02139; alockerd@media.mit.edu

Guy Hoffman (Ph.D. candidate); MIT Media Lab, E15-468a, Cambridge, MA 02139; guy@media.mit.edu

EMERGENCY CARE AND COMPENSATION FOR INJURY

"In the unlikely event of physical injury resulting from participation in this research you may receive medical treatment from the M.I.T. Medical Department, including emergency treatment and follow-up care as needed. Your insurance carrier may be billed for the cost of such treatment. M.I.T. does not provide any other form of compensation for injury. Moreover, in either providing or making such medical care available it does not imply the injury is the fault of the investigator. Further information may be obtained by calling the MIT Insurance and Legal Affairs Office at 1-617-253 2822."

RIGHTS OF RESEARCH SUBJECTS

You are not waiving any legal claims, rights or remedies because of your participation in this research study. If you feel you have been treated unfairly, or you have questions

regarding your rights as a research subject, you may contact the Chairman of the Committee on the Use of Humans as Experimental Subjects, M.I.T., Room E32-335, 77 Massachusetts Ave, Cambridge, MA 02139, phone 1-617-253 6787.

SIGNATURE OF RESEARCH SUBJECT OR LEGAL REPRESENTATIVE

I understand the procedures described above. My questions have been answered to my satisfaction, and I agree to participate in this study. I have been given a copy of this form.

A.1.3 Written instructions given to participants

Thank you for participating in the Game Character Training Experiment. Read these instructions and ask the experimenter if you have any questions.

The Game Setup

In this study you play a video game. (If the application is not yet running on the computer please ask the experimenter to start the application so you can view the game while reading the instructions.) This game has one character, Sophie, a robot in a kitchen. Sophie begins facing the shelf that has various objects that can be picked up, put down, or used on other things (a bowl, a spoon, a tray, flour, and eggs). In the center of the screen is a table, the workspace for preparing foods before they go in the brick oven (on the left hand side of the screen).

Baking a Cake

In this game your goal is for Sophie to bake a cake, but she does not know how to do the task yet. Your job is to get Sophie to learn how to do it by playing this training game. The robot character has 'a mind of its own' and when you press the 'Start' button on the bottom of the screen, Sophie will try to start guessing how to do the task.

Overall steps for baking the cake include:

1. make batter by putting both the flour and eggs in the bowl and
2. mix them with the spoon.
3. then put the batter into the tray
4. then put the tray in the oven

Feedback Messages

You can't tell Sophie what actions to do, and you can't do any actions directly, you're only allowed to give Sophie feedback by using the mouse. When you click the mouse anywhere on the kitchen image, a rectangular box will appear. This box shows the message that you are going to send to Sophie.

- Dragging the mouse UP makes the box more GREEN, a POSITIVE message.
- Dragging the mouse DOWN makes the box more RED, a NEGATIVE message.
- By lifting the mouse button, the message is sent to Sophie, she sees the color and size of the message and it disappears.
- If you click the mouse button down on a specific object, this tells Sophie that your message is about that object. As in, "Hey Sophie, this is what I'm talking about..." (the object lights up to let you know when you're sending an object specific message).
- If you click the mouse button down anywhere else, Sophie assumes that your feedback pertains to everything in general.

Disasters & Goals

Sometimes Sophie will accidentally do actions that lead to the Disaster state. (Like putting the spoon in the oven!) When this happens "Disaster" will flash on the screen, the kitchen gets cleaned up and Sophie starts a new practice round. Additionally, if Sophie successfully bakes the cake, "Goal!" will flash on the screen, the kitchen gets cleaned up and Sophie starts a new practice round. For the disaster state, Sophie is automatically sent a negative message. For the goal state, Sophie is automatically sent a positive message.

Completing the Study

Play the training game with Sophie until you believe that she can get the cake baked all by herself (or you've had enough fun with the training game, whichever happens first!). Note that she may need your help baking the cake more than once before she can do it herself. When you think she's got it, press the 'Finish' button and notify the experimenter. At this point your game character will be tested, and your performance will be calculated based on the time it took you to train the character and how fast your character can bake the cake in a test run.

Practice & Questions

Please take a moment before starting to move the mouse around the kitchen scene and try clicking/dragging the mouse to get used to how you send messages to Sophie. Then tell the experimenter that you are ready to go!

A.1.4 Questionnaire Completed by participants

Thank you for participating in the Game Character Training Experiment. To complete the experiment, we would like you to answer a few questions related to your experience in the game. For each of the statements below, please indicate – on a scale of 1 to 7, the degree to which you agree disagree with the statement. Strongly Disagree is 1, Strongly Agree is 7.

1. My overall experience with the software was enjoyable.
2. I am likely to want to play this game again.
3. The software interface was intuitive and clear.
4. The software interface (not the robot character) was responsive.
5. The robot character was responsive to my commands.
6. The robot character seemed to understand my intentions.
7. The robot character seemed to get better at the task as time went by.
8. The robot character spent much time performing seemingly useless actions.
9. I usually had a good understanding what the robot character was trying to do at a given moment.
10. I usually had a good understanding what the robot character's overarching goals were.
11. When the robot character was making mistakes, I had a good understanding what the root of those errors were.
12. I could generally tell whether the robot character was undecided.
13. My interaction with the robot character had a positive effect on its performance.
14. The robot character understood where I was trying to direct it.
15. The more I invested in teaching the character, the better it became at solving the task.
16. The robot character seemed to have a good sense of what a certain reward pertained to.
17. As time passed, the robot character seemed to need me less and less
18. I have had significant experience with machine learning software and systems in the past.

A.1.5 Informal Interview

After they played the game and completed the questionnaire, each participant talked casually with one experimenter about the experience. They were not prompted with particular questions, just asked to give any thoughts or feedback about the experience.

A.2 Experiment 2 – Online

After the first experiment, we made several modifications to the Sophie's Kitchen game and had a number of hypotheses. The platform was modified slightly to run as a Java applet rather than a Java application. A Webpage was built with an Introductory page, an Informed Consent page, and finally the Java applet. Participants were solicited via MIT mailing lists and advertisements on craigslist. Each participant was randomly assigned to a configuration of the applet that conformed to one of the conditions used for the experiments covered in Chapter 4 and Chapter 5.

In the online version of the game, we had to reduce the task slightly to make the experience shorter. We took away the spoon and the bowl objects. Thus, now to bake the cake Sophie needed to put the eggs and flour in the tray, and then put the tray in the oven. This made the task much shorter, so people were able to spend about 5-15 minutes training Sophie, rather than the 30 minutes needed for the previous experiment. Additionally, the questionnaire portion was conducted through surveymonkey.com directly after they finished training Sophie.

A.2.1 Experimental Protocol

The study protocol is the same as described in Section A.1.1. We received approval to make one modification to the online version of the Sophie experiments regarding payment. We were concerned that if we offered money for the study online we would have people gaming the system by playing many times in order to collect more money. This would bias our results considerably. Thus the IRB board agreed that we could offer the study online without paying people. Since we were asking people to volunteer to play a game, the enjoyment factor is their benefit or compensation. In practice, we found that people did need some motivation to participate. Instead of paying each individual, we had a raffle. Each player had three entries in a raffle for \$100 at Amazon.com.

A.2.2 Introduction page

This is an online game that is part of a research study about how different people try and teach the Sophie agent. Our hope is that people can have fun teaching an agent a simple little task, and that we can learn a little about the teaching process along the way.

How it works:

In this game, players teach the Sophie robot agent to bake a cake. While watching the agent try to bake the cake on her own, players teach by sending various messages via the mouse. The entire activity (playing the game and filling out the survey) takes about 15-20mins. Everyone who completes the study will be entered in a raffle for three chances to win \$100 at Amazon. Please, it is important for the integrity of our study that people only play the game one time. The study has the following steps:

- First you play the game
- When the game comes up, the instructions will tell you about how to use the mouse to communicate with Sophie
- Not everyone has the same instructions about the mouse, so it's important to read these carefully!
- Be sure to practice with the mouse communication before pressing Start because you can't pause the game once it's started.
- When you press the Start button, Sophie will start bumbling around the kitchen trying to bake a cake.
- When you feel like Sophie has learned, press the 'Sophie is Ready' button. Sophie will then try to bake the cake by herself and your score will be calculated based on her success (and how quickly she can do it).
- You then fill out a survey about your experience playing with Sophie, so tell us what you thought!
- Finally, you will be given a link to send us an email to enter the raffle. We will confirm your entry within a day, and the raffle will be run once the study is complete at which time we will notify you with the results. You can only enter the raffle once. We will update this page throughout the study with the number of participants needed, so you can have an idea of when the raffle will happen.

Requirements:

You will need Java 1.4.2 or higher in order to play this game. This website has been tested on the PC with Internet Explorer, and Firefox, and on the Mac with Safari, but if you encounter any problems, please let us know (see the Contact Page).

A.2.3 Informed consent page

After the introductory page the participant is brought to a page that shows the approved consent form seen in Section A.1.2. At the bottom of this page they are asked to click a button 'I agree' if they agree with the terms and wish to volunteer for the study.

A.2.4 Instructions

Once they clicked the 'I agree' button they were taken to the page with the Sophie's Kitchen Java applet. The web-based version of the game has a significantly simpler description of the instructions. They were asked to read all of the instructions and to practice with the mouse interface before pressing the start button. Every player saw the game instructions and the feedback instructions, but only players that were assigned to a condition using the guidance channel of communication saw the guidance instructions.

The Game

This is Sophie's kitchen, she is currently facing the shelf looking at the cooking tools, to her right is a table, and behind her is the brick oven. Sophie needs to learn how to bake a cake. The steps are: Make batter by putting the tray on the table, then add eggs and flour, and finally put the tray in the oven. You can't do any actions for Sophie, or tell her exactly what to do, but you can send messages with the mouse to try and help (details below), Sophie may need help baking the cake a couple of times before she can do it herself, when she can do it, press 'Sophie is Ready!' and she will go into TEST mode, you will get a score based on how many steps she takes and how long you spent training her. After this, please complete the 2 minute survey. Thanks for playing!

Feedback Messages

You can give feedback messages (+/-) after Sophie does an action. When you click the LEFT mouse button a rectangle appears, showing your message for Sophie. Drag the mouse to change the size and color of your message. UP = GREEN (positive), DOWN = RED (negative).

Guidance Messages

You can direct Sophie's attention to particular objects with guidance messages. Click the RIGHT mouse button to make a yellow square (if you only have one mouse button, hold down the option key to do this type of message). Use the square to help guide Sophie

to the right objects at the right times, as in 'Pay attention to this!' Objects light up when the mouse is over them to help you know what guidance message you will send. You can only use the guidance on an object (not a location like the table, shelf or oven). And Sophie only sees your message if she is facing the object. For example, if she is facing the table and you make the yellow square over the flour on the shelf she won't see that, but if you do it when she is facing the shelf, she will see it and think you are telling her to pay attention to or do something with the flour.

A.3 Guidance Experiment – in Lab

The experiment covered in Chapter 3 was conducted in lab. Participants were solicited from the campus community via email and completed the experiment in the lab space of the Robotic Life Group and the MIT Media Lab (E15-468). This experiment used the protocol seen in Section A.1.1, the informed consent seen in Section A.1.2, and the instructions seen in Section A.2.4. In this experiment however, the full set of kitchen objects is used, rather than the reduced set used in the online version of the experiment.

Appendix B

Sphinx Grammar

B.1 Full JSGF Grammar with Parse Tags

```
<numberedTaskNames> = task ( one {TASK-1} |  
    two {TASK-2} | three {TASK-3} | four {TASK-4} | five {TASK-5} | six {TASK-6} |  
    seven {TASK-7} | eight {TASK-8} | nine {TASK-9} | ten {TASK-10} );  
  
<specialTaskNames> = ( <specialTaskNameOn1> |  
    <specialTaskNameOn2> | <specialTaskNameOff1> | <specialTaskNameOff2> );  
  
<taskNames> = <numberedTaskNames> | <specialTaskNames>;  
  
<specialTaskNames> =  
    (turn [all] the buttons ( on {TASK-BUTTONS-ON}| off {TASK-BUTTONS-OFF}) ) |  
    (turn ( on {TASK-BUTTONS-ON}| off {TASK-BUTTONS-OFF}) [all] the buttons ) ;  
  
public <specialTaskNameOn1> =  
    (turn [all] the (buttons | lights ) on) {TASK-BUTTONS-ON};  
public <specialTaskNameOn2> =  
    (turn on [all] the (buttons | lights )) {TASK-BUTTONS-ON};  
public <specialTaskNameOff1> =  
    (turn [all] the (buttons off | lights )) {TASK-BUTTONS-OFF};  
public <specialTaskNameOff2> =  
    (turn off [all] the (buttons | lights ) ) {TASK-BUTTONS-OFF};
```

```

public < GrammarTasks.badSentence> =
( <numberedTaskNames> ) {IMPROPER-PHRASE};

public <question> = (( can you ) | ( could you )) {QUESTION};

public <robotname> = leo | leonardo;

public <greetings> = hello | hi;

public <farewell> = [good] bye | bye;

<actions> = (look at) {LOOK-AT} | point {POINT} |
(where is | show me | find) {FIND} | ( press | push ) {PRESS} |
( flick | flip ) {POINT-FLICK} | ( squeeze) {POINT-SQUEEZE} |
( twist ) {POINT-FLICK-IN} | (double squeeze) {POINT-DOUBLE-SQUEEZE} |
(slide in) {SLIDE-IN} | (slide out) {SLIDE-OUT};

public <affectPos> = (good) | (fun) | (friendly) | (your friend) | (nice);
public <affectNeg> = (bad) | (scary) | (mean) | (not nice);

public <feedback> = (( good job ) | good ) {GOOD-FEEDBACK} |
( not quite | bad ) {BAD-FEEDBACK};
public <fillerPhrases> = the | (at the) | at | to | (to the) | towards;

public <GrammarOther.badSentence> =
( <question> | <robotname> | <greetings> | <actions> ) {IMPROPER-PHRASE};

<numberedButtons> = button (one {BUTTON-1} | two {BUTTON-2} |
three {BUTTON-3} | four {BUTTON-4} | five {BUTTON-5} | six {BUTTON-6} |
seven {BUTTON-7} | eight {BUTTON-8} | nine {BUTTON-9} );

<people> = matt {PERSON-MATT} | jesse {PERSON-JESSE} | marc {PERSON-MARC} |
andrea {PERSON-ANDREA} | cynthia {PERSON-CYNTHIA} | guy {PERSON-GUY} |
zoz {PERSON-ZOZ} | cory {PERSON-CORY} | jeff {PERSON-JEFF} |
dan {PERSON-DAN};

<coloredButtons> = (red {BUTTON-RED} | blue {BUTTON-BLUE} |

```

```

green {BUTTON-GREEN}) button;

<coloredBalls> = (red {BALL-RED} | blue {BALL-BLUE} |
yellow {BALL-YELLOW} | white {BALL-WHITE} ) ball;

<otherToys> = ([yellow] lizard) {TOY-LIZARD} | ( [yellow] fish ) {TOY-FISH} |
( [blue] bucket ) {TOY-BUCKET} | ( elmo ) {TOY-ELMO} | ( kermit ) {TOY-KERMIT} |
( big bird ) {TOY-BIGBIRD} | (where I think elmo is) {HUMAN-BELIEF-TOY-ELMO} |
([toy] box) {TOY-BOX};

<objects> =      <numberedButtons> | <coloredButtons> | <coloredBalls> |
<otherToys> | <people>;

public <GrammarObjects.badSentence> =
( <numberedButtons> | <coloredButtons> | <objects> ) {IMPROPER-PHRASE};

public <BadSentences> = <GrammarObjects.badSentence> |
<GrammarSequences.badSentence> | <GrammarTasks.badSentence> |
<GrammarOther.badSentence> ;

public <questionSentence> =
[<robotname>] [<question>] <actions> [<fillerPhrases>] (<objects> );

public <feedback> = (( good job ) | (good work) | (great job) |
(well done) | (good [<robotname>])) {GOOD-FEEDBACK} |
((not quite) | (try again) ) {BAD-FEEDBACK};

public <labelSentence> =
[<robotname>] (((this is){LABEL} [the] <objects>) | ((my name is){LABEL} <people>));

public <suggestSentence> =
[<robotname>] (try to) {SUGGEST} <actions> [the] <objects>;

public <learnNowTaskName> =
[<robotname>] (((the box is) | (its)) open) {LEARN-NOW OPEN} |

```



```

((((the box is) | (its)) closed) {LEARN-NOW CLOSED}) |
((the box is red) {LEARN-NOW RED}) |
((the box is green) {LEARN-NOW GREEN}) |
((the box is blue) {LEARN-NOW BLUE}) |
((the box is yellow) {LEARN-NOW YELLOW});

public <affectLabelSentence> =
[<robotname>] [the] <objects> is ( <affectPos> {GOOD-FEEDBACK} |
<affectNeg> {BAD-FEEDBACK});

public <commandSentence> = [<robotname>] <actions> [<fillerPhrases>] <objects>;

public <confirmHearSentence> = [<robotname>] (can you hear me) {CONFIRM};

public <greetingSentence> = (<greetings> [<robotname>]){CONFIRM};
public <farewellSentence> = (<farewell> [<robotname>]){FAREWELL};
public <feedbackSentence> = <feedback> [<robotname>];

public <seq1Sentence> =
[<robotname>] [<sequenceWords>] <actions> [<fillerPhrases>] <objects>;
public <seq2Sentence> =
[<robotname>] <actions> [<fillerPhrases>] <objects> [<sequenceWords>];

public <yougo> = (( go ahead ) | ( you [can] go )){YOU-GO};
public <igo> = (( let me [go] ) | ( I [can] go )){I-GO};

public <usgo> = ( ((let us) | ( lets )) do <numberedTaskNames> ) {LET-US-DO};

public <learnTaskSentence> =
[<robotname>] (i will teach you [to] [do]) {LEARN-TASK} <taskNames>;

public <donumTaskSentence> = [<robotname>] do {DO} <numberedTaskNames>;
public <dospecTaskSentence> = [<robotname>] <specialTaskNames> {D000};

public <questionNumTaskSentence> =
[<robotname>] <question> do {D0} <numberedTaskNames> ;

```

```
public <questionSpecTaskSentence> =  
[<robotname>] <question> {D0} <specialTaskNames> ;  
  
public <completedTaskSentence> = <taskNames> (is now done) {DONE};  
  
public <killApp> = ( <robotname> terminate speech recognition) {KILL-SPEECH};
```


Bibliography

- [Ahn and Picard, 2006] Ahn, H. and Picard, R. W. (2006). Affective cognitive learning and decision making: The role of emotions. In *Proceedings of the 18th European Meeting on Cybernetics and Systems Research (EMCSR 2006)*.
- [Argyle et al., 1973] Argyle, M., Ingham, R., and McCallin, M. (1973). The different functions of gaze. *Semiotica*, pages 19–32.
- [Arkin et al., 2003] Arkin, R., Fujita, M., Takagi, T., and Hasegawa, R. (2003). An ethological and emotional basis for human-robot interaction. In *Proceedings of the Conference on Robotics and Autonomous Systems*.
- [Atkeson and Schaal, 1997] Atkeson, C. G. and Schaal, S. (1997). Robot learning from demonstration. In *Proc. 14th International Conference on Machine Learning*, pages 12–20. Morgan Kaufmann.
- [Baldwin and Baird, 2001] Baldwin, D. and Baird, J. (2001). Discerning intentions in dynamic human action. *Trends in Cognitive Sciences*, 5(4):171–178.
- [Bates, 1997] Bates, J. (1997). The role of emotion in believable agents. *Communications of the ACM*, 37(7):122–125.
- [Blumberg et al., 2001] Blumberg, B., Burke, R., Isla, D., Downie, M., and Ivanov, Y. (2001). CreatureSmarts: The art and architecture of a virtual brain. In *Proceedings of the Game Developers Conference*, pages 147–166.
- [Blumberg et al., 2002] Blumberg, B., Downie, M., Ivanov, Y., Berlin, M., Johnson, M., and Tomlinson, B. (2002). Integrated learning for interactive synthetic characters. In *Proceedings of the ACM SIGGRAPH*.
- [Bratman, 1992] Bratman, M. (1992). Shared cooperative activity. *The Philosophical Review*, 101(2):327–341.
- [Breazeal, 2002] Breazeal, C. (2002). *Designing Sociable Robots*. MIT Press, Cambridge, MA.

- [Breazeal et al., 2005a] Breazeal, C., Berlin, M., Brooks, A., Gray, J., Hoffman, G., and Thomaz, A. L. (2005a). Robotic gaze behavior to support human-robot interaction. In *Robotic Life Group - Technical Report 1 (in review at IJCV)*. MIT Media Lab, Cambridge, MA.
- [Breazeal et al., 2005b] Breazeal, C., Berlin, M., Brooks, A., Gray, J., and Thomaz, A. L. (2005b). Using perspective taking to learn from ambiguous demonstrations. *to appear in the Journal of Robotics and Autonomous Systems Special Issue on Robot Programming by Demonstration*.
- [Breazeal et al., 2004a] Breazeal, C., Brooks, A., Gray, J., Hoffman, G., Lieberman, J., Lee, H., Lockerd, A., and Mulanda, D. (2004a). Tutelage and collaboration for humanoid robots. *International Journal of Humanoid Robotics*, 1(2).
- [Breazeal et al., 2004b] Breazeal, C., Hoffman, G., and Lockerd, A. (2004b). Teaching and working with robots as collaboration. In *Proceedings of the AAMAS*.
- [Breazeal et al., 2005c] Breazeal, C., Kidd, C., Thomaz, A. L., Hoffman, G., and Berlin, M. (2005c). Effects of nonverbal communication on efficiency and robustness in human-robot teamwork. In *Proceedings of the IROS*.
- [Breazeal and Scassellati, 2002] Breazeal, C. and Scassellati, B. (2002). Robots that imitate humans. *Trends in Cognitive Science*, 6(11).
- [Brooks and Breazeal, 2006] Brooks, A. and Breazeal, C. (March 2006). Working with robots and objects: Revisiting deictic reference for achieving spatial common ground. In *Proceedings of the 2006 ACM Conference on Human-Robot Interaction (HRI)*.
- [Buchanan and Mitchell, 1978] Buchanan, B. G. and Mitchell, T. M. (1978). Model-directed learning of production rules. In Waterman and Hayes-Roth, editors, *Pattern-directed inference systems*. Academic Press, New York.
- [Burton et al., 1984] Burton, R. R., Brown, J. S., and Fischer, G. (1984). Skiing as a model of instruction. In Rogoff, B. and Lave, J., editors, *Everyday cognition: its development in social context*. Harvard University Press, Cambridge, MA.
- [Clark, 1996] Clark, H. H. (1996). *Using Language*. Cambridge University Press, Cambridge.
- [Clouse and Utgoff, 1992] Clouse, J. and Utgoff, P. (1992). A teaching method for reinforcement learning. In *Proc. of the Ninth International Conf. on Machine Learning (ICML)*, pages 92–101.

- [Cohen and Levesque, 1991] Cohen, P. R. and Levesque, H. J. (1991). Teamwork. *NOÛS*, 35:487–512.
- [Cohen et al., 1990] Cohen, P. R., Levesque, H. J., Nunes, J. H. T., and Oviatt, S. L. (1990). Task-oriented dialogue as a consequence of joint activity. In *Proceedings of the Pacific Rim International Conference on Artificial Intelligence*, Nagoya, Japan.
- [Cohn et al., 2003] Cohn, D., Caruana, R., and McCallum, A. (2003). Semi-supervised clustering with user feedback.
- [Cohn et al., 1995] Cohn, D., Ghahramani, Z., and Jordan, M. (1995). Active learning with statistical models. In Tesauro, G., Touretzky, D., and Alspector, J., editors, *Advances in Neural Information Processing*, volume 7. Morgan Kaufmann.
- [Csibra, 2003] Csibra, G. (2003). Teleological and referential understanding of action in infancy. *Phil. Trans. The Royal Society of London*, 358:447–458.
- [Dennett, 1987] Dennett, D. C. (1987). Three kinds of intentional psychology. In *The Intentional Stance*, chapter 3. MIT Press, Cambridge, MA.
- [Duda et al., 2002] Duda, R., Hart, P., and Stork, D. (2002). *Pattern Classification*. Wiley Interscience.
- [Evans, 2002] Evans, R. (2002). Varieties of learning. In Rabin, S., editor, *AI Game Programming Wisdom*, pages 567–578. Charles River Media, Hingham, MA.
- [Gleissner et al., 2000] Gleissner, B., Meltzoff, A. N., and Bekkering, H. (2000). Children’s coding of human action: cognitive factors influencing imitation in 3-year-olds. *Developmental Science*, 3(4):405–414.
- [Goldman and Mathias, 1996] Goldman, S. A. and Mathias, H. D. (1996). Teaching a smarter learner. *Journal of Computer and System Sciences*, 52(2):255–267.
- [Gray et al., 2005] Gray, J., Hoffman, G., Berlin, M., and Breazeal, C. (2005). Motion generation for expressive interactive robots. In *Robotic Life Group - Technical Report 2*. MIT Media Lab, Cambridge, MA.
- [Greenfield, 1984] Greenfield, P. M. (1984). Theory of the teacher in learning activities of everyday life. In Rogoff, B. and Lave, J., editors, *Everyday cognition: its development in social context*. Harvard University Press, Cambridge, MA.

- [Grosz and Sidner, 1990] Grosz, B. J. and Sidner, C. L. (1990). Plans for discourse. In Cohen, P. R., Morgan, J., and Pollack, M. E., editors, *Intentions in communication*, chapter 20, pages 417–444. MIT Press, Cambridge, MA.
- [Hancher, 2003] Hancher, M. D. (2003). A motor control framework for many-axis interactive robots. Master’s thesis, MIT.
- [Holroyd and Coles, 2002] Holroyd, C. and Coles, M. (2002). The neural basis of human error processing: Reinforcement learning, dopamine, and the error-related negativity. *Psychological Review*, 109(4):679–709.
- [Horvitz et al., 1998] Horvitz, E., Breese, J., Heckerman, D., Hovel, D., and Rommelse, K. (1998). The lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In *In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 256–265, Madison, WI.
- [Isbell et al., 2001] Isbell, C., Shelton, C., Kearns, M., Singh, S., and Stone, P. (2001). Cobot: A social reinforcement learning agent. *5th Intern. Conf. on Autonomous Agents*.
- [Kaelbling et al., 1996] Kaelbling, L. P., Littman, M. L., and Moore, A. P. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285.
- [Kaplan et al., 2002] Kaplan, F., Oudeyer, P.-Y., Kubinyi, E., and Miklosi, A. (2002). Robotic clicker training. *Robotics and Autonomous Systems*, 38(3-4):197–206.
- [Kaye, 1977] Kaye, K. (1977). Infant’s effects upon their mothers’ teaching strategies. In Glidewell, J., editor, *The Social Context of Learning and Development*. Gardner Press, New York.
- [Krauss et al., 1996] Krauss, R. M., Chen, Y., and Chawla, P. (1996). Nonverbal behavior and nonverbal communication: What do conversational hand gestures tell us? In Zanna, M., editor, *Advances in experimental social psychology*, pages 389–450. Tampa: Academic Press.
- [Kuhlmann et al., 2004] Kuhlmann, G., Stone, P., Mooney, R. J., and Shavlik, J. W. (2004). Guiding a reinforcement learner with natural language advice: Initial results in robocup soccer. In *Proceedings of the AAAI-2004 Workshop on Supervisory Control of Learning and Adaptive Systems*, San Jose, CA.
- [Kuniyoshi et al., 1994] Kuniyoshi, Y., Inaba, M., and Inoue, H. (1994). Learning by watching: Extracting reusable task knowledge from visual observation of human performance. *IEEE Transactions on Robotics and Automation*, 10:799–822.

- [L. S. Vygotsky, 1978] L. S. Vygotsky, E. M. C. (1978). *Mind in society: the development of higher psychological processes*. Harvard University Press, Cambridge, MA.
- [Lamere et al., 2003] Lamere, P., Kwok, P., Walker, W., Gouvea, E., Singh, R., Raj, B., and Wolf, P. (2003). Design of the cmu sphinx-4 decoder. In *8th European Conf. on Speech Communication and Technology (EUROSPEECH 2003)*.
- [Lashkari et al., 1994] Lashkari, Y., Metral, M., and Maes, P. (1994). Collaborative Interface Agents. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, volume 1. AAAI Press, Seattle, WA.
- [Lauria et al., 2002] Lauria, S., Bugmann, G., Kyriacou, T., and Klein, E. (2002). Mobile robot programming using natural language. *Robotics and Autonomous Systems*, 38(3-4):171–181.
- [Lave and Wenger, 1991] Lave, J. and Wenger, E. (1991). *Situated learning: legitimate peripheral participation*. Cambridge University Press, Cambridge.
- [Lieberman, 2001] Lieberman, H., editor (2001). *Your Wish is My Command: Programming by Example*. Morgan Kaufmann, San Francisco.
- [Lin, 1992] Lin, L. J. (1992). Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, 8:293–321.
- [Lockerd and Breazeal, 2004] Lockerd, A. and Breazeal, C. (2004). Tutelage and socially guided robot learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
- [Maclin et al., 2005] Maclin, R., Shavlik, J., Torrey, L., Walker, T., and Wild, E. (2005). Giving advice about preferred actions to reinforcement learners via knowledge-based kernel regression. In *Proceedings of the The Twentieth National Conference on Artificial Intelligence (AAAI)*, Pittsburgh, PA.
- [Mataric, 1997] Mataric, M. (1997). Reinforcement learning in the multi-robot domain. *Autonomous Robots*, 4(1):73–83.
- [Morency et al., 2002] Morency, L.-P., Rahimi, A., Checka, N., and Darrell, T. (2002). Fast stereo-based head tracking for interactive environment. In *Int. Conference on Automatic Face and Gesture Recognition*.
- [Nicolescu and Matarić, 2003] Nicolescu, M. N. and Matarić, M. J. (2003). Natural methods for robot task learning: Instructive demonstrations, generalization and practice. In *Proceedings of the 2nd Intl. Conf. AAMAS*, Melbourne, Australia.

- [Oudeyer and Kaplan, 2004] Oudeyer, P.-Y. and Kaplan, F. (2004). Intelligent adaptive curiosity: a source of self-development. In *Proceedings of the 4th International Workshop on Epigenetic Robotics*, volume 117, pages 127–130.
- [Pea, 1993] Pea (1993). Practices of distributed intelligence and designs for education. In Salomon, G., editor, *Distributed cognitions: Psychological and educational considerations*. Cambridge University Press, New York.
- [Peters and Campbell, 2003] Peters, R. A. and Campbell, C. L. (2003). Robonaut task learning through teleoperation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Taipei, Taiwan.
- [Plutchik, 1984] Plutchik, R. (1984). Emotions: A general psychoevolutionary theory. In Sherer, K. and Elkman, P., editors, *Approaches to Emotion*, pages 197–219. Lawrence Erlbaum Associates, New Jersey.
- [Rogoff and Gardner, 1984] Rogoff, B. and Gardner, H. (1984). Adult guidance of cognitive development. In Rogoff, B. and Lave, J., editors, *Everyday cognition: its development in social context*. Harvard University Press, Cambridge, MA.
- [Saksida et al., 1998] Saksida, L. M., Raymond, S. M., and Touretzky, D. S. (1998). Shaping robot behavior using principles from instrumental conditioning. *Robotics and Autonomous Systems*, 22(3/4):231.
- [Schaal, 1999] Schaal, S. (1999). Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3:233–242.
- [Schohn and Cohn, 2000] Schohn, G. and Cohn, D. (2000). Less is more: Active learning with support vector machines. In *Proc. 17th ICML*, pages 839–846. Morgan Kaufmann, San Francisco, CA.
- [Singh et al., 2005] Singh, S., Barto, A. G., and Chentanez, N. (2005). Intrinsically motivated reinforcement learning. In *Proceedings of Advances in Neural Information Processing Systems 17 (NIPS)*.
- [Smart and Kaelbling, 2002] Smart, W. D. and Kaelbling, L. P. (2002). Effective reinforcement learning for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3404–3410.
- [Smith and Scott, 1997] Smith, C. and Scott, H. (1997). A componential approach to the meaning of facial expressions. In *The Psychology of Facial Expression*. Cambridge University Press, United Kingdom.

- [Stanley et al., 2005] Stanley, K. O., Bryant, B. D., and Miikkulainen, R. (2005). Evolving neural network agents in the nero video game. In *Proceedings of IEEE 2005 Symposium on Computational Intelligence and Games (CIG'05)*.
- [Steels and Kaplan, 2001] Steels, L. and Kaplan, F. (2001). Aibo's first words: The social learning of language and meaning. *Evolution of Communication*, 4(1):3–32.
- [Stern et al., 1998] Stern, A., Frank, A., and Resner, B. (1998). Virtual petz (video session): a hybrid approach to creating autonomous, lifelike dogz and catz. In *AGENTS '98: Proceedings of the second international conference on Autonomous agents*, pages 334–335, New York, NY, USA. ACM Press.
- [Sutton et al., 1998] Sutton, R., Precup, D., and Singh, S. (1998). Intra-option learning about temporally abstract actions. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML98)*, Masion, WI.
- [Sutton et al., 1999] Sutton, R., Precup, D., and Singh, S. (1999). Between mdps and semi-mdps: Learning, planning and representing knowledge at multiple temporal scales. *Journal of Artificial Intelligence Research*, 1:11–39.
- [Sutton and Barto, 1998] Sutton, R. S. and Barto, A. G. (1998). *Reinforcement learning: An introduction*. MIT Press, Cambridge, MA.
- [Thomas and Johnson, 1981] Thomas, F. and Johnson, O. (1981). *Disney Animation: The Illusion of Life*. Abbeville Press, New York.
- [Thomaz et al., 2005a] Thomaz, A. L., Berlin, M., and Breazeal, C. (2005a). An embodied computational model of social referencing. In *IEEE International Workshop on Human Robot Interaction (RO-MAN)*.
- [Thomaz and Breazeal, 2006a] Thomaz, A. L. and Breazeal, C. (2006a). Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*.
- [Thomaz and Breazeal, 2006b] Thomaz, A. L. and Breazeal, C. (2006b). Transparency and socially guided machine learning. In *Proceedings of the 5th International Conference on Developmental Learning (ICDL)*.
- [Thomaz et al., 2005b] Thomaz, A. L., Hoffman, G., and Breazeal, C. (2005b). Real-time interactive reinforcement learning for robots. In *AAAI 2005 Workshop on Human Comprehensible Machine Learning*.

- [Thomaz et al., 2006] Thomaz, A. L., Hoffman, G., and Breazeal, C. (2006). Experiments in socially guided machine learning: Understanding how humans teach. In *Proceedings of the 1st Annual Conference on Human-Robot Interaction (HRI)*.
- [Thrun, 2002] Thrun, S. (2002). Robotics. In Russell, S. and Norvig, P., editors, *Artificial Intelligence: A Modern Approach (2nd edition)*. Prentice Hall.
- [Thrun and Mitchell, 1993] Thrun, S. B. and Mitchell, T. M. (1993). Lifelong robot learning. Technical Report IAI-TR-93-7.
- [Trevvarthen, 1979] Trevvarthen, C. (1979). Communication and cooperation in early infancy: A description of primary intersubjectivity. In Bullowa, M., editor, *Before Speech: The Beginning of Interpersonal Communication*, pages 389–450. Cambridge University Press, Cambridge.
- [Voyles and Khosla, 1998] Voyles, R. and Khosla, P. (1998). A multi-agent system for programming robotic agents by human demonstration. In *Proceedings of AI and Manufacturing Research Planning Workshop*.
- [Watkins and Dayan, 1992] Watkins, C. and Dayan, P. (1992). Q-learning. *Machine Learning*, 8(3):279–292.
- [Wertsch et al., 1984] Wertsch, J. V., Minick, N., and Arns, F. J. (1984). Creation of context in joint problem solving. In Rogoff, B. and Lave, J., editors, *Everyday cognition: its development in social context*. Harvard University Press, Cambridge, MA.
- [Woodward et al., 2001] Woodward, A. L., Sommerville, J. A., and Guajardo, J. J. (2001). How infants make sense of intentional actions. In Malle, B., Moses, L., and Baldwin, D., editors, *Intentions and Intentionality: Foundations of Social Cognition*, chapter 7, pages 149–169. MIT Press, Cambridge, MA.
- [Zalla et al., 2000] Zalla, T., Koechlin, E., Pietrini, P., Basso, F., Aquino, P., Sirigu, A., and Grafman, J. (2000). Differential amygdala responses to winning and losing: a functional magnetic resonance imaging study in humans. *European Journal of Neuroscience*, 12:1764–1770.
- [Zukow-Goldring et al., 2002] Zukow-Goldring, P., Arbib, M. A., and Oztop, E. (2002). Language and the mirror system: A perception/action based approach to cognitive development. *Working draft*.