UNIVERSITY OF CALIFORNIA SAN DIEGO

Assistive Healthcare Robotics and Preference Learning

A Thesis submitted in partial satisfaction of the
requirements for the degree of Master of Science

in

Computer Science

by

Bryce Woodworth

Committee in charge:

Professor Laurel Riek, Chair
Professor Henrik Christensen
Professor Ryan Kastner

2020

The Thesis of Bryce Woodworth is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

Chair

University of California San Diego

2020

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Thank you to Fracesco Ferrari, Teofilo E. Zosa, and Professor Laurel D. Riek, whose collaboration made this work possible. Thanks also to all the members of the UC San Diego Healthcare Robotics Lab, whose advice, encouragement, and feedback have proven invaluable.

The authors would also like to thank Michelle Yu, PT, for her assistance in the development of the prehabilitation routine, and Monique F. Narboneta for designing the study robot's interface layout in addition to the algorithm overview figures.

Finally, thank you to my friends and family, for bearing with me through this process and making it all worthwhile.

Chapters 4 and 5, in part, have been published as it appears in "Preference Learning in Assisitive Robotics: Observational Repeated Inverse Reinforcement Learning", Machine Learning for Healthcare Conference 2018 [102]. The thesis author was the primary investigator and author of this paper.

ABSTRACT OF THE THESIS

Assistive Healthcare Robotics and Preference Learning

by

Bryce Woodworth

Master of Science in Computer Science

University of California San Diego, 2020

Professor Laurel Riek, Chair

Robots have the potential to revolutionize the way healthcare is delivered, particularly with regard to improving access to it. In-home robotic assistants could be used to extend care in a consistent, scalable, and personalized way. To realize this vision, we decompose the requirements of such a system, propose a target architecture, and illustrate where more research effort is needed. In particular, we focus on preference inference as a subproblem in which we aim to make direct progress.

Current preference learning techniques lack the ability to infer long-term, task-independent preferences in realistic, interactive, incomplete-information settings. To address this gap, we introduce a novel preference-inference formulation, inspired by assistive robotics applications, in

which a robot must infer these kinds of preferences based only on observing the users behavior in various tasks. We then propose a candidate inference algorithm based on maximum-margin methods, and evaluate its performance in the context of robot-assisted prehabilitation. We find that the algorithm learns to predict aspects of the users behavior as it is given more data, and that it shows strong convergence properties after a small number of iterations. This result moves us towards the vision of more helpful and personal in-home robotic assistants, and demonstrates the tractability of future progress in this area.

Chapter 1

Introduction

Amidst the many rapidly advancing fronts in technology, robotics is among the forerunners in terms of its potential to impact the world and improve quality of life [19, 64, 43]. Potential uses for advanced robotic systems include driverless cars, surgical systems, personal assistants, and many more [19]. As technological improvements are made, robots are becoming more capable across a wide range of fields.

One such field is occupational therapy. The American Occupational Therapy Association defines occupational therapy as "the only profession that helps people across the lifespan to do the things they want and need to do through the therapeutic use of daily activities" [98]. Occupational therapists assist clients in a wide variety of manners, including teaching children how to properly hold pencils, helping those with traumatic brain injury to remember important information, and ensuring that older adults interact with their home environment in a manner that is safe for them [51]. Unfortunately, the demand for healthcare-related services such as occupational therapy is larger than the supply of available healthcare providers [88]. Though there are many potential future applications of robotics, in this work we will focus on concrete problems through the lens of an occupational therapy (OT) assistant.

While it is impossible to provide every person in need with assistive living or a full-time

occupational assistant, the existence of robust and accessible robotic assistants could provide many of them with access to higher-quality care. For example, a robot could help carry groceries for someone with a physical impairment, give medication reminders to someone with memory-related difficulties, or demonstrate proper form for OT-specified stretches and exercises while providing valuable feedback. The physical nature of robots also allows them to provide users with social interaction, which has been shown to be particularly valuable to patients with dementia [94]. In addition, such robots could enhance the safety of their users by providing diagnostic data, notifying caregivers or the OT of any pertinent changes in physical or mental health.

Despite these potential applications, robots have historically found success primarily in environments in which they are physically isolated from humans, such as industrial assembly lines [11]. Recently, there has been a trend towards desiring more and more autonomous systems that work around and interact with humans. However, the kinds of designs and programming that allowed robots to find success on factory floors do not extend well to robots that need to work closely with humans. In particular, such systems need to have an understanding of the humans they are working with in order to be safe around them and successfully perform interactive tasks in dynamic environments [80].

One large bottleneck in building interactive robots is that such systems will need to be adaptive, meaning they personalize their interaction to individual users over time. Humans differ in capabilities and preferences in nuanced and often dramatic ways, and an interactive strategy that works well for one human may be ineffective or even counterproductive for another [41]. In addition, humans have a strong intuitive notion of individual idiosyncrasies, and this understanding is crucial for our interactions and our ability to coordinate on problem solving [23]. In order for robots to provide engaging and socially interactive behavior, they must have an understanding of such personalized adaptation. While this is true of all sufficiently interactive robots, it is especially critical in the realm of occupational therapy, in which we expect to see high variance in the capabilities and preferences of the users as well as high sensitivity to unsafe

or counterproductive behaviors caused by misunderstandings.

Currently there are several commercially successful (but non-robotic) products that serve as personal assistants. These include personalities embedded in mobile devices as well as those built into immobile standalone hardware [10]. Examples of these assistants include: Apple's SIRI, Microsoft's Cortana, Google's Google Now, and Amazon's Alexa. However, while these products can successfully service a limited range of user requests, they are not sufficient to meet broader needs, such as those outlined above for occupational therapy. In particular, such systems are immobile, cannot give physical demonstrations of tasks, cannot carry out physical tasks, and are usually not equipped with the kinds of sensors we want for observing and interacting with their human counterparts. Jeong et al. provide further empirical evidence that physical robotic systems are better suited to providing healthcare support than virtual characters [49].

There has been some research towards extending assistant technology to the realm of personalized robots that can meet the wide range of potential tasks demanded by applications such as occupational therapy. For example, Saunders et al. [85] developed a home assistant robot that allowed the user to build up a set of actions to be taken when logical conditionals were met, and add condition-action sets in real-time. Kidd and Breazeal [50] demonstrated a robotic weight-loss coach with limited adaptability based on responses to survey questions. Others created a therapeutic game system for stroke patients based on the Microsoft Kinect, which adapted to users through the use of a calibration phase to infer the user's range of motion [55].

Despite the successes of these assistive systems, many are still insufficient for tackling the applications in adaptive robotic healthcare we are interested in. In particular, there is a need for systems amenable to programming by non-experts to perform a wide variety of tasks. Many of these tasks will not be easy to specified by logical conditionals on simple environmental states, and a more comprehensive input system is needed. Additionally, we desire a robust mechanism for user-adaptation that functions well under real-world constraints. The system should not require the user to manually specify all of their activities and preferences, should understand that

preferences can shift over time, and should work in domains for which no simple calibration exercise exists.

1.1 Contributions

The contributions of this work are as follows:

- **Providing a roadmap:** We present a more complete roadmap that specifies the open challenges to building a usable in-home robotic assistant, particularly in occupational therapy.
- **Defining an architecture:** We provide architectural proposals for solving the problems laid out in the above roadmap, and successively decompose and analyze the components that do not yet exist to expose gaps in understanding.
- **Formalizing adaptivity:** We narrow in on a subproblem in adaptivity and user-modelling, and introduce a formal framework to model it.
- **Proposing solutions:** We describe why existing work does not satisfactorily solve this adaptivity framework, and introduce algorithmic solutions.
- **Empirical validation:** We test one of our proposals in a real-world occupational therapy environment with a physical robot and human users, demonstrating the efficacy of our approach.

1.2 Thesis overview

The rest of the work is organized as follows:

- Chapter 2 explores relevant existing work and how it ties into the goals of an adaptive, in-home, OT assistant.

- Chapter 3 further solidifies the design goals, proposing and motivating an architecture to give more concrete steps on what work needs to be done.
- Chapter 4 formalizes the preference inference problem that we are attempting to address, and introduces solutions.
- Chapter 5 discusses empirical experiments that test the efficacy of one such inference algorithm.
- Chapter 6 concludes and indicates directions for future work.

1.3 Publications

Some of the work presented in this thesis, particularly the text in Chapters 4 and 5 and Background sections 2.3 and 2.7, was published as a paper in collaboration with Francesco Ferrari, Teofilo E. Zosa, and Laurel D. Riek:

Woodworth, B., Ferrari, F., Zosa, T., Riek, L.D. (2018) "Preference Learning in Assistive Robotics: Observational Repeated Inverse Reinforcement Learning" In Proceedings of Machine Learning for Healthcare (MHLC). [Acceptance Rate: 19%] [102].

Chapter 2

Background

This chapter discusses existing work relevant to the construction of assistive robotic systems. Section 2.1 gives an overview of assistive robotic approaches. Section 2.2 discusses reinforcement learning, the primary formalism for agents that interact with their environments. This is extended to inverse reinforcement learning in section 2.3, which studies agents that reason about other agents. Active learning explores systems that can choose which questions to ask for more clarification on, and is discussed in section 2.4. Section 2.5 discusses existing work in understanding human workflows. Further discussion of a concrete application area is provided in section 2.6, which discusses automatic tutoring systems. Section 2.7 covers existing work on systems that can understand the difference between global preferences and task-dependent preferences. Finally, section 2.8 discusses important problems in robustness and safety as they relate to these systems.

2.1 Overview of Assistive Robotics

While there is not currently a unified approach that meets the demands of an adaptable robotic assistant, there is existing work that has tackled narrower versions of the problem. Examples of this include the embedded personal assistants discussed in Chapter 1. In addition

to this, there is some existing work explicitly on home robotics. For example, Saunders et al. demonstrate a robotic assistant integrated into a smart home filled with relevant sensors [85]. The assistant has two user-input modes for adaptability: "teaching", which involves specifying a series of if - then conditions, and "showing", which involves presenting a new label and demonstrating the activity it represents. While successful in its application domain, the architecture only understands the state based on specific home sensors, and its adaptability formulation requires new understanding or behavior to be explicitly input by the user. We desire to work towards systems that can operate successfully in any home, and which do not require the user to be willing or able to directly teach it how to operate.

Kidd and Breazeal demonstrate a different kind of robotic assistant that focuses on long-term social interaction in the context of weight loss [50]. The robot primarily uses a text-based conversational system that encourages the user to log their diet and exercise, and walks them through the logging process with small talk. After a verbose introductory phase, it calculates a binary relationship status by asking the user questions each day, which modifies dialogue. The authors found that having a system that had varied conversational structure and eyes that appeared to look at the user led to greater success (measured in affinity and actual weight loss) than pen-and-paper logging. In the context of occupational therapy, these results lend credence to the hypothesis that users will be better served by personalized, social robotic interaction than by, for example, receiving reminders on a smartphone.

Another approach that explicitly targets healthcare is presented by Lange et al. in the context of physical rehabilitation [55]. Their system aids in rehabilitation when close personal interaction with a therapist is infeasible due to a shortage of rehabilitation specialists. They gamify the motions used in the rehabilitation by creating a simple gem-collecting game displayed on a screen, with the user interacting through a Microsoft Kinect. The system can adapt to users through an initial calibration phase that asks the user to perform a motion and uses this to calculate their range of motion, placing the gems accordingly.

There has also been work on directly training or assisting healthcare professionals. Moosaei et al. [67] build a robotic patient simulator with facial pain perception, for use in training clinicians, while Moharana et al. [65] introduce a paradigm in which robots are trained to assist informal dementia caregivers. Taylor et al. [96] target an application with robot-assisted coordination of clinical teams of nurses.

Iqbal et al. [44][45] explore a method of adaptivity for human-robot teaming. They demonstrate algorithms that allow a robotic system to understand temporal patterns in human activity and adapt accordingly. This was validated in the context of a human dance routine that a robot would dynamically understand and join. Further work by Iqbal and Riek extends this to the case of a collaborative human-robot drumming scenario [46] and human-robot handovers [47].

For a further survey of existing applications of robotics in rehabilitation, see Luxton and Riek [61]. As a more forward-looking take, Lee and Riek perform a study on the currently-unmet potential for robots to promote successful aging [56].

While these approaches successfully address various aspects of the full problem, we will have to build on them with work from many other disciplines in order to create a robot for robust and adaptable in-home occupational therapy assistance.

2.2 Reinforcement Learning

Reinforcement learning is a learning paradigm in which an agent learns through interactions with the environment that lead to rewards [91]. In a given timestep, the agent selects an action to take in the environment. Based on this action, as well as the previous history of everything in the environment, the environment will provide the agent with a reward signal and a set of observations. For tractability, environments are generally modelled such that the relevant parts of their history is encapsulated in a single "state" variable. Altogether, in this framework the environment has a state, an agent chooses an action to perform, and this causes the environment

to update to a new state and provide the agent with observations and a reward.

The goal of a reinforcement learning agent is to maximize the cumulative (discounted) reward it achieves in the environment. Doing this generally requires a combination of *exploration*, in which the agent takes unfamiliar actions in order to learn about the environment, as well as *exploitation*, in which the agent takes actions it knows to be good in order to maximize reward. Reinforcement learning is a particularly promising framework for acting in complex environments, as it is generally easier to specify some general reward function over states than it is to specify whether an action is "correct" or not as in supervised learning.

There is an inherent trade-off between conservative reward functions, which reward only the states in which the goals are satisfied, and more aggressive reward functions that reward apparently promising intermediary states. The former runs into problems with tractability, as the agent must explore huge swathes of uninformative states before ever reaching a state with positive reward. On the other hand, the latter can encourage the agent to perform the wrong task altogether as it collects rewards from non-goal states [91]. This is discussed in more detail in section 2.8 on Robustness and Safety.

There are a variety of potential techniques that can provide more information to the agent about the desirableness of states, while still encouraging the proper behavior. For example, Jaderberg et al. [48] propose an agent that looks for pseudo reward signals in the environment that are good predictors for the eventual reward. This allows the agent to use environmental heuristics - such as score, location, or items - to indicate which intermediate states are more or less promising than others. With this modification, the authors show a 10x reduction in training time with a 3% (Atari) to 50% (3D labyrinth) improvement in score over a standard deep learner (A3C LSTM) without the environmental indicators.

Another option is to leverage human expertise by incorporating implicit information from a human observer as part of the reward function. As one example, Broekens motivates the usage of affective facial expressions as an augmentation to the reward signal [21]. They argue for an

agent that can estimate the valence associated with a human observer's facial expression, which may provide crucial additional data about the agent's performance. Since humans rely heavily on inferred facial affect in daily social interactions [29], we may expect that robots will also benefit from this information. Broekens further demonstrates an agent which uses an estimate of facial affect as an additional reward parameter. This agent is able to learn faster in a simplified setting than an agent which does not take affect into account, providing empirical credence to the usefulness of incorporating data from human observers.

An implicit system (such as one involving inferred facial affect) has the advantage of being a natural reward signal in cases where the goal is maximizing a human's preferences. In many nontrivial tasks, a human may not be willing or able to provide explicit per-action feedback to an agent, and in this case implicit signals are the only option. However, it is inefficient to rely only on implicit feedback when other alternatives are available.

Alternatively, human evaluators may specify the reward function explicitly. Thomaz et al. [99] explore what kinds of reward signals non-expert humans provide to a robot when they are attempting to teach the robot how to perform a task. They found that humans tend to give rewards not just based on whether the current state satisfies the goal, but also based on predicted future actions; they attempt to use the reward signal to guide the robot. Another finding was that humans build up intuitive models of the underlying learning algorithm, which they use to adapt their reward signal over time. Although having humans explicitly provide all reward signals can lead to robust and self-correcting behavior, it is infeasible to have a human observe and evaluate every action taken during training and deployment of an agent.

2.3 Inverse Reinforcement Learning

Inverse reinforcement learning (IRL) is the problem of attempting to infer the reward function being optimized by an external agent [68]. Once the reward function has been learned,

one can then use a standard reinforcement learning algorithm to choose a course of action that maximizes the reward in the current context. The modelling of agent motivations is in contrast to behavior cloning, another approach to learning from another agent. In behavior cloning, the goal is to directly infer a policy that most closely matches the observed behavior [100].

Unlike behavior cloning, inverse reinforcement learning agents attempt to explicitly model the reward function of the other agent. That is, instead of attempting to find a policy that mimics the other agent's actions, the goal is to find the reward function that led to those actions. If the inferred reward function is accurate, the agent can better generalize to environments that are dissimilar to the ones for which it has expert demonstrations. However, IRL assumes that the external agent is behaving optimally with respect to its reward, which does not hold for humans [84]. In addition, it suffers from problems of underspecification, as there are infinite reward functions that could lead to a particular set of observed actions [68]. For example, the uniform reward function causes every possible trajectory to be optimal, and thus will always be included in the hypothesis set.

The combination of suboptimal demonstrations and underspecification pose a serious problem for IRL. Because of underspecification, any given behavior will be optimal with respect to many potential reward functions. With suboptimal demonstrations the true reward function will likely not even be in that set of candidates. Ng and Russell attempt to address this problem by adding in heuristics that favor some reward hypotheses over others [68]. These include a penalty on overly-general reward functions that are satisfied by many observations, as well as a penalty on reward functions that highly reward too many states. These heuristics encourage the inferred reward function to be both simple and specific, and can make the optimality constraints soft so that the agent will favor a simple and specific but slightly suboptimal hypothesis over a vague, general, and optimal one.

One can further expand the practicality of IRL by allowing an agent to learn from failed demonstrations as well as successful ones. As humans regularly learn from experiences of failure

(e.g. in the realm of mathematics, see Tall [93]), the intuition behind this approach is to allow robots that same affordance. This allows the robot to learn not only about a narrow set of cases involving success, but also about a variety of potential failure modes to avoid. Thus they can directly rule out overly-general reward functions that predict success everywhere, such as the uniform reward function, as it fails to predict that the observed demonstrations constitute failure.

A more specific formulation is cooperative inverse reinforcement learning (CIRL) [40]. In this setting, a human (H) and robot (R) are in a reinforcement learning setting where both agents are rewarded according to H's reward function, but R does not initially know what H's reward function is. This way H makes a tradeoff between directly maximizing the reward themselves and taking actions which more effectively communicate the reward to R so that R can render more effective assistance, depending on what they expect to maximize their own reward. This formulation has the benefit of treating the human as part of the optimization process rather than just part of the environment, allowing it to model how humans can adapt to the robot's behavior while teaching them. However, the downside of this is that optimal behavior requires the human to understand the robot's algorithm and current beliefs, and select actions to maximize the robot's information. For a non-expert human who only knows how to teach a task by demonstrating it, CIRL reduces to standard IRL.

Inverse reinforcement learning techniques are often used to teach robots to perform human-like motions. In this context it is commonly referred to as inverse optimal control, which addresses the same problem but from the formalism of optimal control [57]. Work that uses these tools has successfully taught robots to grasp objects [37], backflip [25], and walk [27]. While successful, work along this front tends to focus on lower levels of abstraction, such as fine-grained motor movements. More work is needed to extend this work to larger environments with complex objectives requiring high-level planning.

2.4 Active Learning

Active learning is the branch of machine learning that deals with giving the learner unlabelled data and letting it choose which data points will have their true label revealed; See Settles for a review [86]. This paradigm is primarily useful when one has a large quantity of unlabelled data at our disposal, but evaluating the label for any given data point is computationally expensive. In the the field of robotics, a robot may quickly generate many potential plans or trajectories for satisfying a given goal. Since in this case it is generally not straightforward to evaluate the relative promise of a trajectory without directly interacting with the environment, one can generate hypothesis trajectories much faster than one can evaluate them, leading to the necessity of carefully selecting which trajectories to prioritize.

In many cases a human will be the one providing feedback to the agent. As discussed in Section 2.2 on reinforcement learning, it is infeasible to expect a human to manually provide feedback for all potential data points. Active learning ameliorates this by significantly reducing the amount of labelling that the human has to perform, instead relying on clever choices of which data to label.

Reinforcement learning can be augmented with the ideas from active learning in a couple of ways. For example, Epshteyn et al. [33] propose active learning as a mechanism to handle possibly misspecified transition probabilities. In their system, the expert-specified environmental dynamics are not used to directly optimize the reward, but are instead used as a blueprint for autonomous exploration to determine the true dynamics through observation. Sensitivity analysis is employed to prioritize exploration, so that actions whose dynamics the reward function is most sensitive to are sampled first.

Another option is to treat the reward function itself as uncertain, and then use active learning to improve the agent's understanding of the reward. This is particularly relevant when the reward function is based off of some human's difficult-to-formalize intuitive goals. Additionally,

in many cases such goals are vastly simpler to identify than to directly demonstrate, meaning approaches such as IRL and imitation learning are insufficient. This can happen for a variety of reasons, notably when the human is a non-expert or when the robot's morphology is sufficiently different from the human's that trajectories cannot be easily mapped between them.

One example of such a system was designed by Zhang [107]. In that work, the authors propose a design in which a human is presented with groups of trajectories, and is asked to select the trajectory from each group that best satisfies their goals. The robot selects a group of policies that approximately maximizes the expected value of information of the query, such that knowing the answer to the query maximally increases the expected return. Thus, like Epshteyn et al., they maximize the expected *impact* of the query, rather than the expected *information* gained from it. High uncertainty about an irrelevant part of the reward function is not given significant weight.

Christiano et al. extend this with a similarly motivated approach that involves learning to perform a task solely by selectively querying a human for preferences over a pair of trajectories [25]. In this architecture, a deep neural network is used to maintain an estimate of the reward function, which is updated through asynchronous interaction with the human, while another reinforcement learner learns a policy based on the estimated reward function. Their approach queries the human on less than 1% of interactions with the environment, and successfully learns a variety of atari and simulated robotic manipulation tasks. Their system performs as well or better than a reinforcement learner given access to the predefined underlying rewards. It also has the benefit of robust self-correction, as opposed to predefined heuristics encoded in reward functions which can lead to optimizing the wrong behavior [91].

Active learning techniques have also shown promise in robotic contexts. For example, Cohn et al. [28] demonstrate active learning techniques for learning the dynamics of a robot arm. Ribes et al.[79] use active learning in a robotic musician attempting to learn how to imitate musical patterns. Additionally, Gordon and Breazeal [39] apply these techniques in building a social robot for tutoring children in word-reading skills.

King et al. [53, 52] extend active learning beyond the context of getting answers from a human. They propose a "robot scientist" capable of performing experiments related to gene functioning in yeast. The robot comes up with hypotheses based on its current state of knowledge, and uses active learning techniques to choose which hypothesis to test in a robotic lab environment. This experiment selection process is shown to be competitive with human performance, and significantly improves upon simple heuristics.

2.5 Workflow Detection

When performing complex, long-term interactions with a human, robots will need some mechanism for understanding what the human is doing and estimating their next actions. One approach towards addressing this is activity detection [3], which seeks to understand human activities based on sensor readings. Particularly relevant to the current work is a subset of activity detection known as workflow detection. In workflow detection, a "workflow" is a structured temporal pattern of tasks (or phases), and the goal of workflow detection is to take as input the workflow topology and a series of observations and output the phase of the workflow that each observation represents [73]. A successful application of workflow detection would allow the robot to be better equipped to understand not only the activity they are performing right now, but also the larger pattern that it fits within and how this relates to the human's goals. Such an understanding would enable the robot to interact more appropriately and to better aid the user.

A recurring problem in workflow detection is building up a robust state representation in the face of occlusion and other object tracking issues. Padoy et al. [73] address this by foregoing tracking of individuals altogether in favor of a global state descriptor. This approach was found to be successful in the context of an operating room surgery, as focus was always centered around the patient.

In other contexts, such a global state descriptor is insufficient, and alternative solutions

are needed. Arbab-Zavar et al. [8] propose using overhead cameras to avoid issues of occlusion, allowing them to provide a localized workflow model that can detect multiple concurrent workflows for different individuals. Alternatively, Stauder et al. [90] detect workflows using only instrument usage data, circumventing the entire issue of detection at the cost of additional necessary sensor data. As discussed in Section 2.1, Saunders et al. [85] show this technique can work in an assistive robotics context, as they relied on sensor data from a smart home filled with sensors.

Most approaches to workflow detection use hidden markov models (HMMs), or some variant thereof, as the tool for analyzing time-series observation data, which requires the specification of well-behaved states and features. Since this can be difficult to specify in complex environments, Cadene et al. [24] propose the use of a residual neural network architecture. This has the benefit of discovering its own state representations from pixel data, rather than requiring expert-defined states and features. However, as is generally the case with neural networks, it is difficult to discern what aspects of the input image caused the network to output a certain phase, making it difficult to understand and to debug.

The particular goal of in-home robotic assistance presents its own challenges and opportunities as it relates to workflow detection. First, while smart devices are becoming more ubiquitous, many homes will likely continue to be devoid of sensors, and we desire our OT assistant to be operable in surch circumstances. Thus unlike [90] and [85], we cannot rely wholly on external sensors to handle our state formulation, though they may be a useful addition when available. On the other hand, such robots are often expected to interact with only a single user at a time, making tracking and detection easier. Face-detection algorithms can distinguish between differing users, which can be easily trained in a calibration phase while the user is being introduced to the robot. In single-user cases the robot may be able to identify the user simply by what house they are in. It is important to take these unique detection considerations into account when designing an in-home robotic assistant.

While there are many promising existing approaches, more work needs to be done before they are robust and generalizable enough to rely on for our robot's interactions, which are subject to very high expectations of safety and reliability. The surgical and industrial settings studied to date consist of very well-defined and consistent tasks. Some home tasks may have similar characteristics, but many involve a more dynamic and less well-defined environment. As one example, a cleaning task may vary greatly from user to user, as each user performs different types of activities in their workflow, with different tools, in different orders. The robot cannot assume the user will always follow a single pre-defined workflow topology in these cases. A more robust and flexible workflow detection solution would greatly enhance the interactive capabilities of robots, and even partial solutions may provide benefits as long as their shortcomings are well understood.

2.6 Automatic Tutoring Systems

A primary goal in human-robot interaction is learning about users over time. One specific domain in which this problem has been addressed is automatic tutoring systems, which attempt to provide a personalized curriculum that most effectively teaches a subject to a human user. An integral component of this goal is inferring the current level of understanding of users as well as how they respond to various types of information presentation, so the system can select and present information in the best way to teach each individual user. Recently, emphasis has been placed on inferring the user's learning style, based on the empirical stability of this trait and trends in the psychology literature [5].

One difficulty in preference inference that occurs when inferring learning styles is the difficulty of getting ground-truth data to update the models. If we do have access to reliable ground-truth information in the form of training data, we can use supervised learning approaches, such as the neural networks used by Bernard et al. [17]. In their case, they used the output of

a well-validated learning style questionnaire to provide the feedback for their model. Another approach, advanced by Dorça et al. [30], is to evaluate the performance of students through testing in order to determine how performance correlates to the assumed learning style. In their study, they evaluated a reinforcement learning model that received rewards based on how well the user empirically demonstrated understanding.

Gordon and Breazeal [39] demonstrate the usefulness of inferring a student's current level of ability, rather than their learning style. They created a robotic system that encourages the student to create a story and help "teach" the robot how to read. The robot selects one word out of a set to present to the child, using techniques in active learning as discussed in Section 2.4. The system is able to accurately infer the student's reading ability, and use this knowledge to create a personalized tutoring approach that yields more uniform results across age and initial-ability ranges than an alternative, randomized approach.

Regardless of approach, there are two components in personalizing to a specific user in an adaptive system. The first of these is user-modelling. In the case of inferring the user's learning style correctly, many approaches achieve accuracies of 70-80%, measured by taking survey results as ground-truth [17]. As inferring learning style typically just requires selecting or ranking 4-7 potential categories, this level of performance is lower than we would like for a robust human-compatible assistant. While imperfect, these methods are likely sufficient to enable significant improvements over non-adaptive systems.

The second component of personalization lies in incorporating the user model in a way that better satisfies the goals of that particular user. There are currently "no proven recipes" for applying inferred learning styles to adaptive tutoring systems, with few success stories of applications in real-world tutoring [22]. This holds despite the fact that tutoring is a reasonably well-studied subset of the full set of domains in which artificially intelligent systems may assist humans. That is, there is pre-existing expert-defined information about learning styles, such as bug libraries — collections of well-known learning roadblocks — that can be taken into account

in the models [103].

These two steps (which will likely be profitably intertwined) are necessary in any adaptive system. More work needs to be done both in improving the performance of automatic tutoring systems as well as in extending these results to the domain of robotic assistants, in which we cannot rely as heavily upon existing expert models for all interaction. The difficulties encountered in tutoring, particularly in usefully applying inferred learning styles, is likely indicative of further difficulty in adaptively utilizing a more comprehensive user model. Still, the results of this field are cause for optimism that we can infer reliable user models just from observed behavior. Through analogy to tutoring, user modelling should be the first area of focus for building an adaptive robotic assistant, as there is likely more low-hanging fruit and better models will improve our ability to evaluate the performance of a method of utilizing such models.

2.7 Inferring Global Preferences

Normally when we talk about preferences in the context of machine learning, we talk about preferences over policies (or preferences over trajectories, from which we can infer preferences over policies). This leads to the problem of inferring an optimal (or pareto optimal) policy through stated preferences between trajectories, in a similar vein to some of the work discussed in the section on active learning. There is much research geared towards answering this problem [25, 20, 106], but this work is generally in the context of what we shall call *local* preferences; that is, preferences over policies in the context of a single well-defined task. A somewhat distinct goal is to infer a *global* model of the user's preferences.

In the context of interactive robotics, we desire our system to model the user in such a way that it can personalize many interactions to the user over a long period of time. As such, we are trying to model task-independent properties of the user that distinguishes them from other users, rather than just what they want to see in this given context. Such a user-specific, task-independent

model is what we refer as a global model of the user. A desideratum of a long-term adaptive system is to build up global models of the user in terms of capabilities (what they can do) and preferences (how they would like to be interacted with, and how that influences the perceived successfulness of an interaction). For now, we focus on the problem of inferring preferences, which is discussed more starting in Section 3.5.

Some of the work done on inferring local preferences for given tasks may also be relevant for inferring global preferences. For example, formalizations of preferences and their potential relations to utility functions, such as those provided by Wirth and Fürnkranz [101], can also be applied to global preferences. They describe a distinction between preference-learning systems that rely on explicitly modelling the preference relation over policies, and systems that indirectly model this relation through inferring an underlying utility function (which is a non-stationary reward-like function that maps policies to numeric utility). They also discuss methods of translating between preferences over trajectories, which we can actually query users for, and preferences over policies, which is what we ultimately want since the goal is to define a policy that will work with any initial environment. While motivated in the context of inferring local preferences, the methods are just as applicable to modelling global preferences, given the ability to differentiate between task-specific preferences and task-independent preferences.

A good example of how to do this kind of differentiation is provided by Amin et al. [6] with repeated inverse reinforcement learning (RIRL), in which we apply IRL over many distinct tasks. In this framework the agent knows all of the task-specific rewards, but we assume that the human's true reward function (which determines their behavior) is the sum of this task-specific reward function and a user-specific reward function shared among all tasks. Such a task-independent reward function can be used as the formalization of a user's global preferences.

The goal of their work is to infer and imitate the behavior of a human expert. In this formulation the expert observes the robot's actions and notifies it if any action is sufficiently suboptimal according to the human's true reward. The expert then provides an optimal demonstration from

the state in question, allowing the robot to update its estimate of the human's task-independent reward function. The main result of their work is an updating procedure for the task-independent reward function, which has provable bounds on the number of times the human will have to intervene, given assumptions about the correctness of their formalism.

There is good reason to expect this task-independent term in a human's effective reward function. In their paper, Amin et al. motivate this by noting that in all tasks that a human performs, they may act according to intrinsic goals about health, security, adherence to law, etc., even when this is not directly part of the task they are undertaking. Russell and Norvig [83] also point out this consideration in the context of a robot planning a path: if the fastest way to get somewhere involves knocking over a vase, we don't want the robot to take that path, even though we didn't specify ahead of time that "vase-safety" is part of the reward for path planning. These preferences are likely shared by most humans, but there are also task-independent preferences that may differ widely between specific humans, such as what kinds of music they enjoy or whether they are bothered by receiving reminders while reading.

One notable related result is that, although we can provably bound the number of times the robot will surprise the human, in the general case we cannot make any guarantees that we will learn the human's task-independent reward to any particular degree of fidelity. Rather than a shortcoming of the approach, this is simply due to the fact that in settings where we cannot arbitrarily select tasks for the human to evaluate, we may never encounter a task that disambiguates the reward for a given state. This is not a problem in the context of having the robot perform tasks according to the human's preferences, as the only way we will be unable to decide between two potential reward functions is if no task we encounter is noticeably affected by the difference between them. However, in the context of healthcare robotics, we may wish to infer effective preferences for diagnostic reasons, to provide relevant updates on physical and mental health. If this is the case, we may propose suitably informative tasks for the human to perform, and Amin et al. prove that, if the user is cooperative, we can recover the actual task-independent

reward in a number of tasks logarithmic in the desired precision.

While amenable to provable guarantees, the procedure by which the human is assumed to reliably intervene and demonstrate optimal trajectories from a given state is unrealistic in the healthcare domains we are interested in. In our case, the task the robot is trying to perform may at least be partially specified by external experts, such as prompting an interactive session in which the robot teaches the human a new exercise. The user is not expected to be expert enough to teach the robot how to teach them the exercise, and is not expected to be observing the robot closely enough to be able to detect the point of divergence into suboptimal behavior even if they are familiar with the task the robot is performing. Thus, while a useful candidate formalism for global preferences, this approach is not by itself sufficient to solve the global preference inference problem in our domain.

A separate idea for building user models is to incorporate results from psychology, similar to the way pre-existing expert knowledge was incorporated into intelligent tutoring systems. For example, Evans et al. [34, 35] build models that incorporates knowledge of several kinds of human biases and suboptimalities due to bounded reasoning, such as time inconsistencies and incorrect beliefs. This knowledge allows the system to gain a better understanding of a human's actions, as it is now able to learn whether surprising behavior is actually caused by the human's preferences being different than expected, or if it is simply an artifact of such psychological effects.

As motivation, Evans et al. provide the example of a human trying to walk to and eat at a healthy restaurant. Even though the human wants to eat at the healthy restaurant, if they pass by a donut shop on the way, they may get tempted and eat at the donut shop instead. Alternatively, if the human is aware of this property, they may go out of their way and take a longer path to avoid passing by donut stores. Such behavior will likely lead to confusion for a standard IRL agent observing the human, who will get conflicting evidence on whether the human actually prefers to eat at the healthy restaurant or the donut shop. This can be avoided if the agent has a more

psychologically realistic user model.

There are clear benefits to providing our robot with this kind of additional data; in particular, the models have the benefit of being both explicit and easily interpretable. This is clearly beneficial in the context of healthcare, in which we desire both that our robots are safe and reliable, as well as able to provide relevant diagnostic data to medical personnel. The current trend in ML is towards less easily-interpretable architectures, such as deep neural networks, which can have unforeseen safety problems [74] and whose internal representations can be difficult for humans to interpret (though work is being done on this front [105]).

The cost of such a system is that it requires explicit, psychologically realistic models in order to function. While it would be useful to extend this psychological modelling approach to the full preference inference problem by having an explicit model of human preferences, this is infeasible with the current state of psychological knowledge. Even if an explicit model of all preferences is currently infeasible, our systems may benefit from the addition of specific, well-researched psychological models, in a similar vein to Evans et al.'s work on bounded agents. With a hybrid approach it may be possible to utilize the representational power of models such as deep neural networks, while maintaining some degree of interpretability from explicit psychological models.

2.8 Robustness and Safety

When designing robots to act in healthcare scenarios, it is vital to ensure that such systems behave safely and robustly in a wide variety of scenarios. Some of the challenges in this area can be addressed through the design of the robot itself, for example by constructing it out of soft materials that minimize the damage caused in collisions [62]. However, other challenges appear at the level of the robot's reasoning and decision-making systems, and cannot be readily addressed through hardware design choices. In this work we are primarily interested in designing

reasoning modules for such robotic systems, so we will focus on the latter case.

In the past, intelligent agents such as robots have relied heavily on expert-specified policies. While there were still notable concerns about systems-level security [76], such robots did not run into many reasoning-based safety issues, as all possible behavior was more directly specified by the human expert. However, as application domains become more and more complex, agents will need to do more and more of their own autonomous reasoning and planning. For this kind of autonomous robot, new classes of safety concerns become relevant, which will become ever more important as the reasoning capabilities of artificial systems improves.

There are a wide variety of outstanding safety-oriented challenges relevant to this context. Many such problems have been laid out in various research agendas, including those proposed by Amodei et al. [7], by Taylor et al. [97], and by Soares and Fallenstein [89]. Problems in reasoning can occur in various places in the design, training, and deployment of goal-directed agents. These include biased training distributions, poor reward specification, mismatches between intended and effective reward signals, aversion to modification, and vulnerability to adversaries.

One area of difficulty lies in the data the agent is given to train from, namely in ensuring it is sufficiently representative of the kind of data the agent will encounter in the environment [7, 97]. This can cause problems when the agent encounters new environments and attempts to extrapolate a solution from insufficient information. One notable historical example of this is a classifier trained to detect whether a picture of a forest contained tanks hidden in the trees, which initially performed admirably, but failed spectacularly when new data was presented [31]. The system evaluators realized that all the pictures of empty forests were taken on a cloudy day, and all pictures with tanks were taken on a sunny day, leading the classifier to simply look at the overall brightness of the image. When presented with pictures of tanks on cloudy days or empty forests on sunny days, the classifier did not have sufficient data to make the correct generalization (is there a tank), and thus extrapolated upon the wrong information (is it sunny).

There are a few approaches to addressing this issue of mismatched training data. Active

learning can address this issue in the context of imitation learning, through the use of DAGGER, a method of avoiding compounding errors that happen when any perturbation causes the state to deviate in a way not present in the training set [82]. DAGGER and related methods mitigate this problem by asking for the optimal responses to the previously-unseen states, thus expanding the training set. In this setting, we simply aggregate all of the environmental states that our agent has experienced and ask for demonstrations involving those states. This requires unrealistic levels of human interaction, but more sophisticated active learning approaches exist which have strong theoretical bounds on the number of required queries. In one such approach, the agent maintains a set of candidate hypotheses that roughly fit the observed data, and asks for a datapoint to be labelled based on the disparity between the predictions of these hypotheses on that datapoint [18].

Even if our agent has access to good training data, poor choices of reward functions can lead to unanticipated and harmful actions. Clark and Amodei [26] describe a case in which a reinforcement-learning agent that was trained to play a racing game learns to ignore human-desired goals such as "win the race", achieving higher reward by driving in a circle and repeatedly hitting point-generating targets. Although it was clear to the authors what the goal of the game was, and there was an intuitive link between rewarding score and rewarding winning races, this still led to surprising and counterproductive behavior. Such behavior may be harmless in the context of a game, but it would be unacceptable in medical contexts.

This problem is further exacerbated by reward shaping, in which the reward function is modified to make it easier to learn. Consider the problem of teaching a robot how to play chess with no prior knowledge. A natural reward function would be to give the agent a score of +1 for being in a winning state and -1 for being in a losing state. However, if the robot initially plays random moves in an attempt to learn the game, against any even remotely skilled opponent it will almost always lose and observe an uninformative reward of -1, making training of such an agent computationally infeasible.

Reward shaping addresses this by providing additional information about the game

through the reward function. In the context of our chess robot, this may involve giving positive reward when the robot captures the opponent's pieces and negative reward when it loses its own pieces. This makes the game significantly easier for the agent to learn as it gets more informative feedback, but the robot is no longer optimizing the same reward function.

As Sutton and Barto [91] point out, the robot may learn to play a strategy that its human designers would not approve of. For example, it may learn to capture opposing pieces and protect its own even at the cost of losing the game. They use this example to motivate the notion that it is improper to use the reward function to describe *how* to achieve the goals, with the reward function's only correct usage being to describe what the goals are. While an important notion, it may be overly pessimistic: Ng et al. [69] describe a necessary and sufficient condition for applying reward shaping in a manner that does not alter the optimal policy.

Another more subtle manner in which a robot can learn incorrect behavior is if the reward function is computed in terms of proxies for real-world objects. This can incentivize the robot to optimize over the proxies rather than the actual objects they are supposed to refer to, a phenomenon known as wireheading [97].

More concretely, consider the case where our goal is to design a robot that learns how to get the user to take the correct medication at the correct time. We desire to reward such a robot based on whether the user successfully took the correct medication, but in our autonomous setting there is no overseer directly providing this reward. Thus we may instead reward the robot if its *internal* state estimation puts it in a state where the user has taken the correct medication. However, this means that if the robot can manipulate its state estimation, for example by damaging its sensors in a specific manner of swapping out all of the labels on the medication bottles to claim they are the desired medication, then it can achieve high reward by essentially tricking the reward function into believing it had accomplished the goal. In cases where this is easier or more reliable than actually maximizing the intended reward (for example if the user always takes random medication regardless of the robot's efforts) then the robot will be incentivized to perform

such wireheading by default.

Everitt and Hutter [36] propose a modification of reinforcement learning called value reinforcement learning (VRL) to address the wireheading problem. In VRL, there is assumed to be some underlying utility function, and the observed reward is based upon this but can be modified if the agent is in a self-delusional state. The goal is not to maximize reward, but to learn and maximize the true utility function, which incentivizes against self-deception as this interferes with the agent's ability to learn the utility function. Everitt and Hutter then define a consistency-preserving VRL agent, and prove that such an agent does not wirehead.

As we have discussed, it can be difficult to define a goal for the agent that causes it to do what we want. Thus, we would like the agent's goals to be easily modifiable if it turns out it behaves in undesirable ways. However, the agent will almost assuredly have instrumental incentives to avoid being modified in this way, and an agent intelligent enough to understand this will actively work against having its goals modified [15]. Intuitively, this is because nearly any goal will be better satisfied if the robot is pursuing that goal than it would if the robot's goals were changed, and the robot's current actions are decided entirely on the basis of what will maximize its current goals.

A common lens through which to view this problem is in the willingness of an agent to let itself be turned off by human overseers. Hadfield-Menell et al. [40] analyze a game-theoretic scenario in which the robot can choose to disable its off-switch before the human overseer has a chance to press it. Standard reinforcement learning agents are incentivized to disable the off-switch in this scenario, but the authors show that it is possible to do better. If an agent is uncertain about its own reward and believes that the human knows its reward better than it itself does, it will be incentivized to allow the human to turn it off while simultaneously avoiding incentives to turn itself off. An example of such an agent can be found in cooperative inverse reinforcement learning, in which the robot's reward function is identical to the human's, but only the human knows what the reward is [40]. Complementary work is done by Orseau and

Armstrong, [72] who define a framework in which an agent can learn the optimal policy even in the face of interruptions.

All of these challenges stem from inherent difficulties in correctly specifying goals, but the problem becomes even more difficult in the presence of adversaries. As an example of this, Behzadan and Munir [14] demonstrate a vulnerability in a common class of deep reinforcement learning agents. For such agents, an attacker with modest abilities to influence the environment can manipulate the agent into adopting the attacker's own policy.

Issues with adversaries are further exacerbated by concerns for privacy. Robotic assistants may have access to sensitive personal data about their users, which a malicious user should not be able to recover through interaction with the robots [92]. One way this can manifest itself is in standard security vulnerabilities, in which an attacker manipulates the system into performing undesired behavior such as divulging private data. However, this is not the only way an attacker can extract information; results related to differential privacy have shown that even exposing aggregated statistical data about a large group can reveal personal data about individuals within the group [32]. If the behavior of a robot depends upon statistical data about its users - for example through the use of a learned prior over preferences - then care must be taken to avoid exposing sensitive user data.

While these concerns need to be addressed when designing modules for robots that act in safety-critical domains such as in healthcare, this is not a complete specification of the manners in which a robot's reasoning system may be unsafe. Notably, many problems in reasoning may depend on the learning architecture being used, and may not materialize in other domains or even in earlier versions of the same system. As a concrete example of this, deep Q-networks, which are commonly used in reinforcement learning tasks, suffer from a problem that causes them to gradually "forget" about previous experiences that they learned to avoid [59]. Agents built from such systems may periodically repeat past mistakes, no matter how catastrophic.

As there is currently no comprehensive checklist for creating robustly safe reasoning

for autonomous agents, it is vital that designers of safety-critical robotic systems place a heavy emphasis on ensuring the reliability of their systems.

Chapter 3

The Robotic Occupational Therapy Assistant (RobOTA): Goals and Architecture

The goal of creating an adaptive in-home robotic occupational therapy assistant is well motivated. The world faces an aging population, most of whom would prefer to age gracefully and independently in their own homes [63]. Unfortunately, high costs, health disparities, and a shortage of healthcare providers make it infeasible to provide this type of care to everyone [81, 75, 13]. In-home systems to provide occupational therapy could help address this care gap, and, if designed well, could support care receivers, providers, and informal caregivers [81].

While the need for such a system is well-motivated, how it is embodied within technology remains an open question. For example, what kinds of behaviors would such a robot need to support, how might one evaluate the degree to which a particular system satisfies the relevant therapeutic goals, how to best interface with clinicians to support end-user goals, etc. Even when such behavioral goals are clearly stated, it is not immediately clear what research effort is necessary to achieve them or how to incorporate disparate research results into a unified

system. In this chapter, we clarify the goals of the project to create an adaptive, in-home Robotic Occupational Therapy Assistant (RobOTA), and characterize the additional research work that is necessary to build such a system.

The rest of the chapter is organized as follows. In Section 3.1, we discuss the design goals a robotic OT assistant should satisfy, based upon our work with OT experts. We then propose RobOTA's architecture in Section 3.2, and discuss how such a design will satisfy our goals. Section 3.2.1 discusses the user-context awareness component in more detail, while Section 3.2.2 concludes the architectural section with a discussion of relevant open problems.

We begin discussion of adaptivity in Section 3.3, which is the problem area we focus on in future chapters. Section 3.4 then discusses important subproblems within adaptivity. In Section 3.5 we present a framework for user modelling, and we extend this to a formal specification of the preference inference problem in Section 3.6. This formal framework is used to propose and evaluate solutions to user modelling in the following chapters. Thus in the course of this chapter we start from the overall design goals, and use them to narrow in on a single formal problem statement that is amenable to direct work.

3.1 Defining the Goals of the System

Our intuitive, overarching goal is to design an adaptive in-home robotic OT assistant (RobOTA). As discussed in Chapter 1, existing robotic systems are insufficient for this purpose. In this section, we will go into more detail on what new kinds of behaviors we desire such a robot to exhibit, and set up criteria for success. This will enable us to make more concrete proposals and evaluate their potential.

When deploying a system in home environments, it is important a robot fit within the clinical context between the care receiver and OT expert, and is cognizant of the OT's workflow in order to be accepted and used [38, 81]. In our desired workflow, an occupational therapist (OT)

Table 3.1: High-level tasks a RobOTA should be able to aid with, along with examples.

Task	**Example**
Social Engagement	Facilitated conversation
Memory	Medication reminder
Encouragement	Support for scheduling events
Acute oversight	Notification of medical personnel after a fall
Long-term oversight	Reporting health changes to OT
Manipulation	Pick item up off the floor

evaluates the care receiver in clinic and co-designs a set of goals with them. The OT can then transfer these goals to the robot. The robot will then later interact with the care receiver in their home in order to aid them in fulfilling the OT-specified goals.

Thus, this can be framed with our robot operating in two phases: specification and deployment. In the specification phase, the OT, who is not assumed to be familiar with robotics or programming, will specify their desired behavior to the robot. This can include specifying low-level activities, such as the motion involved for a particular exercise, as well as high-level procedures such as the initiation and guidance of an interactive session that teaches the care receiver how to perform the exercise while providing useful feedback.

In the deployment phase, the robot will interact with the care receiver in the home to assist them in satisfying the OT's goals. In particular, the system will need to be able to understand the relevant aspects of the user and environment, as well as the goals, in order to decide when it is appropriate to perform various actions. Given that different users are expected to behave in different ways, and change at different rates throughout their rehabilitation, the robot should have mechanisms that allow it to learn about the user over time, such that it can personalize its behavior to better suit their individual needs. This is what we refer to as *adaptivity*.

Through a series of contextual interviews with occupational therapists (see Chapter 5), we identified several important target activities to design RobOTA around. These tasks were deemed

to be both helpful for those needing care, as well as technologically feasible by todays robotic systems. These tasks are decomposed with examples in Table 3.1.

Some cognitive tasks that a successful RobOTA should assist in include social engagement, memory, and encouragement. In social engagement and reorientation, the user is assisted through interaction that includes providing reorienting data such as the user's name and the current date [95]. RoboTA may also assist users by providing reminders, such as for medication, including identification of the correct kind. Users may additionally benefit from encouragement to plan events, along with general support for scheduling them.

RobOTA could combine these social signals with physical sensing to provide both physical and psychological oversight. This includes monitoring for acute health conditions, such as falls, as well as more long-term monitoring for early signs of declines in health. Finally, many users would greatly benefit from direct physical assistance with mobility and the movement of household objects.

Our goal for RobOTA's architecture is for it to be flexible enough to eventually support all of these goals in a manner that is as widely-applicable as possible. Thus, we make as few assumptions as we can about, for example, how familiar the OT is with robotics, the kinds of actions we may be asked to take, the properties of the user or their home, etc. In particular, we desire such a system to be extensible enough to enable additional forms of behavior as new environments are encountered and improvements in technology enable new hardware and software capabilities for the robot.

In line with all of this information, we consider a design for this robot to satisfy our goals if it can:

- Expose a simple, easy-to-use declarative interface to occupational therapists
- Be expressive enough to handle any of the kinds of action or regimen OTs may use
- Be flexible enough to handle unforeseen capability and environment changes

- Successfully implement the regimen with the user such that it fulfills the OT's goals

3.2 Proposed RobOTA Architecture

Figure 3.1 shows a diagram for a proposed high-level architecture for addressing these goals. We have specified this architecture at a high enough level of abstraction that it can apply to a wide range of robotic system implementations, while at a concrete enough level to clearly identify open research questions of interest.

Descriptions corresponding to the numbered labels on the diagram are as follows:

1. **Occupational therapist (OT)**: works with the user and determines a set of goals for RobOTA.

   (a) During the specification phase, the occupational therapist designs a set of desired goals for the robot to assist the user in, which gets compiled into behavior nodes in an execution graph.

   (b) In some cases when the therapist describes how to perform new behavior, some additional information may be provided to help the robot understand what kinds of properties this behavior has.

2. **Option Models**: decompose potential actions the robot could take into a set of relelvant features describing those actions.

3. **Environment**: includes everything that RobOTA will be interacting with during deployment, including the user.

4. **Sensors**: on (or off of) the robot perceive the environment, and constantly publish large streams of low-level data.

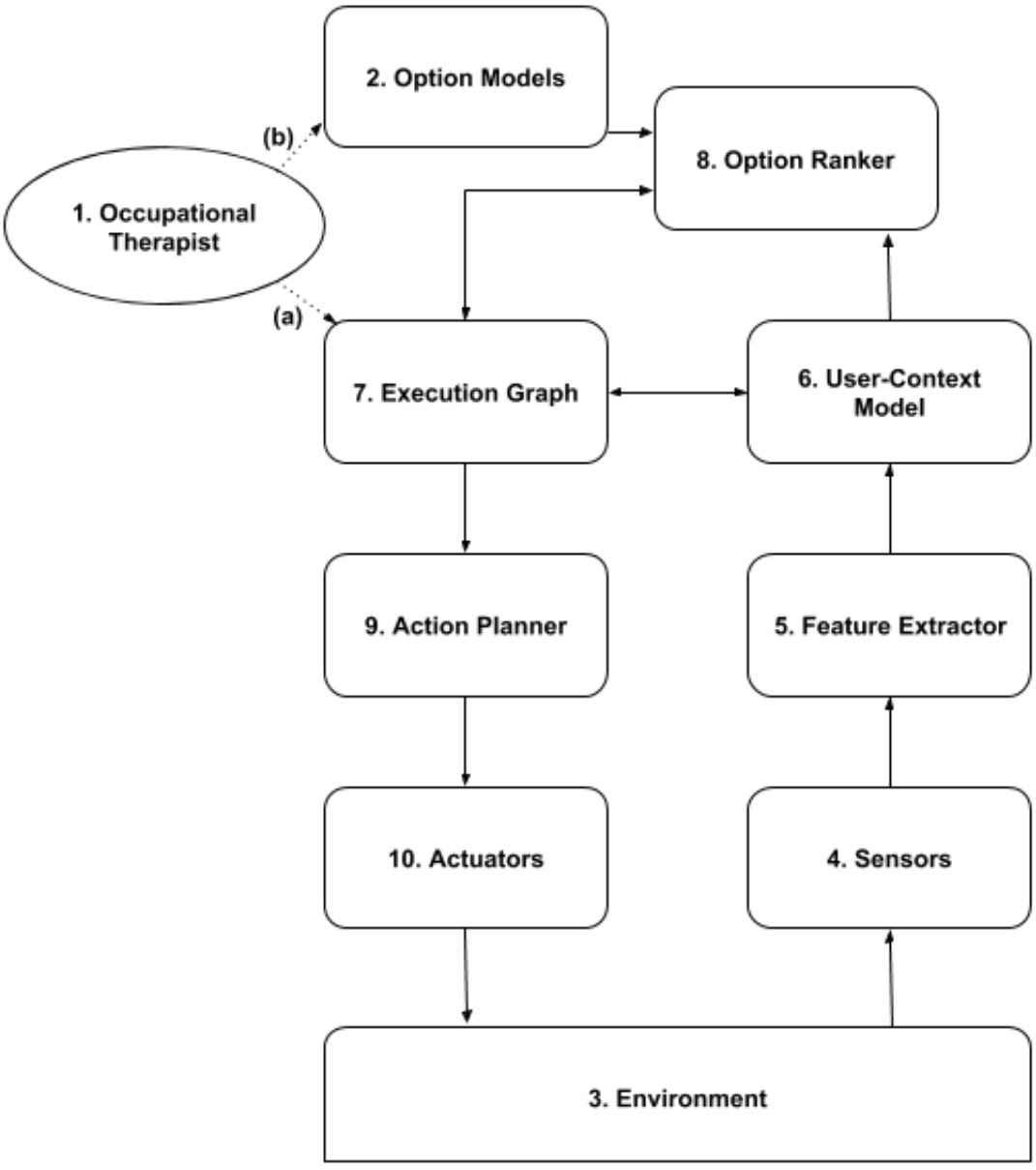


Figure 3.1: A high-level architecture for an adaptive in-home robotic occupational therapy assistant. Dotted lines represent passing of information during the specification phase, while solid lines correspond to the deployment phase

5. **Feature Extractor**: makes this stream of low-level data useful for higher-level reasoning by putting it through a feature extraction step, where the large quantity of relatively unstructured data is transformed into a simple and coherent environmental model. For example, this node might include a skeleton-tracking algorithm.

6. **User-Context Model**: uses this higher-level data to make inferences about the user (capabilities and preferences), as well as the context (location, history, workflow, etc).

7. **Execution Graph**: keeps track of which goals are currently unmet, and decides which execution node, if any, it would be appropriate to schedule right now. To do this, it sends the data about which options it is considering to the option ranker, which responds with scores for each alternative based on how appropriate it is in the current context. Additionally, the execution graph may need to take additional contextual data into account for various types of execution nodes; in this case it queries the user-context model directly.

8. **Option Ranker**: monitors the current user-context model for features which are relevant to determining the appropriateness of the options. When the Execution graph queries the Option Ranker for a set of options, the relevant option features are extracted through the Option Models, and combined with the contextual data to return a ranking.

9. **Action Planner**: takes as input the action that the execution graph has selected, uses the appropriate handlers to decompose the action into primitive operations, and schedules the actions for physical execution.

10. **Actuators**: take signals from the action planner and physically execute them in the environment.

Such a design would allow us to express and implement any of the expected tasks specified above, as well as to have the flexibility to extend to a wide range of other tasks. This is largely because the therapist-specification procedure is intentionally underdefined. That is,

concrete systems for that purpose can be modularly substituted without invalidating the rest of the subsystems, as long as they can compile the goal information into a form usable in a decision-making node (the execution graph). For example, some parts of this specification could happen in manners similar to those described by Saunders et al. [85].

While all of the aspects of this architecture are important and require additional research, in this thesis we scope the work to focus on those related to adaptivity and action selection. For action selection, the execution graph will have an understanding of the goals specified by the therapist, and which ones are currently unmet. We explicitly model the action-selection decision in terms of how appropriate various actions related to goal completion are. This is accomplished by combining models of the user, the context, and the actions themselves, with adaptivity being dependent upon the user-model and how it is incorporated into decision-making. These pieces of information are all necessary in order to make the best decisions, as removing them would render the system incapable of understanding the environment, the user, or itself.

This data enables the robot to understand the environment, the user as an individual, and how it can impact both. We expect any successfully adaptive architecture to incorporate the kinds of information discussed in this model. Thus, when we use this information to discuss the kinds of challenges faced, we can be confident that such problems and analysis will be relevant to future concrete implementations of a fully-operational RobOTA.

3.2.1 The User-Context Model

The user-context model abstracts away sensor data from the decision-making module (the execution graph), exposing just the details that are necessary to make good decisions. It exposes a model that encapsulates all the necessary data about the individual properties of the user and the current state of the user and environment.

It is also a fundamental building block of adaptivity, as it contains longer-term models of the user. These can be used both by caregivers, for example, by the presentation of a report of a

user's changes in physical capability, as well as by the system itself, as it allows alternatives to be chosen based on an individual user's inferred preferences.

As a concrete example of how such a model enables adaptivity, consider the case of a system that has two users. Alice is hard of hearing, whereas Bob is made anxious by sudden loud noises. In this case, when giving verbal reminders we will want to be louder when working with Alice and quieter when working with Bob. When working with Bob, the system may additionally learn to prefix all sound-based actions with a soft attention-getting chime that gently ramps up in volume. For both users, the system may opt to give information visually more often than verbally.

Internally, there are three distinct (though not orthogonal) inferences taking place: preference inference, capabilities inference, and context-awareness. *Preference inference* refers to an attempt to understand the users intentions, goals, and desires. *Capabilities inference* refers to an attempt to understand what types of actions the user is willing and able to perform. *Context-awareness* refers to an attempt to understand the context in which the user and robot are operating, and how that interacts with their goals and actions.

These different inferences have different interpretability requirements. For capability inference, models should be human-interpretable, in order to generate reports to the OT as needed. While models for preference inference and context-awareness may also benefit from being interpretable, this is not a requirement. As such, the features of those two models may be able to be learned in an opaque way, such as through a deep-learning system. If these features are learned, they should be learned in tandem with the option ranking system, which will use the features as input.

There are many potential directions we could take in tackling these inference problems, using for example techniques in learning theory, but some open questions remain.

3.2.2 Relevant Open Questions

The design work done thus far brings to light a few core research questions that will have to be answered in order to actually build a system that functions as desired. For example, how can we make the therapist-facing interface as simple and intuitive as possible while retaining as much expressive power as possible? In simple cases OTs may desire to interact entirely verbally through natural language processing. However, in more complex cases there are many difficulties in making such an NLP system expressive enough to understand the range of concepts the therapist may wish to specify.

One way to make specification easier for the therapist is to create classes of related actions and goals, new instances of which can be quickly parameterized to fit the current need. For example, we may have a "programming by demonstration" class [4, 9, 54] that allows a therapist to record a series of motions that can then be translated into a robot's actuators for later performance. How can we build and incorporate such a set of core action classes and goal classes? It is possible to explicitly program in such classes ahead of time if we can predict the need for them, but can we allow the therapist to have a framework for easily creating new classes as unforeseen needs arise?

Many kinds of behavior, such as social conversation, will likely be relevant for all users. Rather than having the therapist repeatedly specify all such behavior, it should be transparently embedded into the execution graph for each user. What kinds of behavior should be included in this set, and how can we ensure that the therapist has all the relevant information about what behavior is being added? How can we ensure that, in unusual contexts that require more fine-grained control, the therapist is presented with the tools and knowledge they need to exert such control over this transparently-added behavior?

The option ranker subsystem has to incorporate information from the user, the context, and the options under consideration in order to perform its task. How should this information be represented, such that it is expressive and easy to work with for the option ranker? The options

we will be deciding between can be single actions or the start of more complex activities. How should we design the features for options that give all relevant information, including information about how the decision will affect the future context? Once the ranker has all of this information, how should it best incorporate it into a unified ranking?

The user-context model is responsible for abstracting all higher-level sensor features into a unified and queryable model. What kind of models of the user and context would capture all of the information that the system needs, while maintaining accuracy under a diverse range of scenarios? If this should be an online reinforcement learner parameterized by user engagement and adherence to therapist-specified goals, how do we extract a usable feedback signal from our observations such that we can successfully update our model? Could we leverage a supervised approach, for example by having a training set of example scenarios along with human expert ratings of the appropriateness of various responses? If so, how could we procure reliable training data that effectively captures the huge range of behavior our system may potentially encounter?

While all of these questions will have to be addressed, in the scope of the current work we will direct focus towards developing methods of adaptivity.

3.3 Adaptivity

There is a spectrum of autonomy that we could support. One end of this might involve an execution graph that is entirely specified in advance by an expert. On the opposite end of the spectrum, we might have the expert specify only the end goals, and have the robot learn how to act in the environment to satisfy those goals through reinforcement learning. We refer to approaches near the former end of the spectrum as "constrained", and approaches near the latter end as "unconstrained".

A constrained system has higher requirements on the expert to design ahead of time all of the interactions and alternatives that might be relevant. Furthermore, the robot can only execute

behaviors that have been explicitly added in as an option by the expert, reducing flexibility to new scenarios. In the worst case, involving many different specified behaviors that could be relevant at the same time, the option selection procedure may require as much user, context, and action feature inference as the unconstrained case. The primary benefits of such an approach are that it is simpler to actually implement and run in the deployment phase, and we have much higher robustness and safety confidence in the system as it does not run into the reasoning issues discussed in Section 2.8.

By comparison, an unconstrained system would be very easy for the expert to program, as they just need a way to specify a goal function. Such a system also allows a significantly higher degree of flexibility and capacity to adapt to new scenarios. However, it requires a significantly higher degree of reinforcement learning horsepower in order to function properly in such complex real-world environments, which the current state of the art seems unable to supply. An unconstrained system likely requires all the user, context, and action modelling that the constrained case requires, plus a bunch of additional work in autonomous reinforcement learning capabilities. This becomes especially challenging considering the aforementioned difficulties in safe autonomous decisionmaking.

It is important to realize that this is not a binary question between constrained and unconstrained systems, but rather a spectrum. For example, completely constrained systems may be infeasible for addressing the range of contexts we are interested in, but mostly-constrained systems that use some learning and build up libraries of known tasks and actions may be sufficient. Through characterizing both ends of the spectrum, we can focus our efforts on the open questions relevant to a wide range of agents, rather than questions that will only benefit a smaller range of the spectrum. As such, in this work we will focus our efforts on the kinds of modelling problems that both kinds of systems need in order to operate, and leave for future work questions about, among other things, how to improve the capabilities of real-world reinforcement learning systems.

3.4 Subproblems in Adaptivity

With this in mind, the following set of capabilities seems sufficient to enable a flexible adaptivity framework:

1. A method of turning percept history into expressive state features that encode all known, relevant contextual information

2. A method of understanding potential actions the robot could take

3. A method of inferring user-specific preferences and capabilities and distilling them into a useful model

4. A method of combining the above three components into an action-selection framework that allows the propagation of training signals based on observed outcomes

We need to ensure the state features are simultaneously expressive enough that they encode all of the data we are interested in, while also being compressed enough that they can be stored and worked with in a computationally efficient manner. Some examples of the kinds of information the state features need to encapsulate includes the locations of the user and robot, the user's mental state, what actions the user is performing, what workflow the user is taking part in, etc. These features could be learned from previous interactions, defined beforehand by an expert, or some combination thereof. We may want our understanding of state to be extensible so we can more easily adapt to new scenarios, such as if the robot will be expected to model a level of danger posed by the use of various household objects for a particular class of user. This will be harder with more opaque approaches such as learned features in deep neural networks.

For action features, we want to be able to understand the actions we can take in terms of the context and user, including how they lead into future workflows such as interactive sessions. As a concrete example, all actions that play a sound or lead to actions that will play a sound should share some feature, such that those actions can be penalized when the user is doing something

sound-sensitive such as talking on the phone. One idea for addressing this is in a model-based approach, in which we assume that there is some transition function that operates on the state features, such that each action corresponds directly to some change (or distribution over changes) in the state features. A "model-free" alternative might involve simply giving actions their own set of representative features (similar to states), and using this directly as input into the function which combines inputs into option rankings, which will have to internally learn how the action features correspond to the environment.

In terms of modelling preferences, we must determine both how to represent the model and how to update the model based on observational feedback. One option is for the model to be expressed in an explicit manner, through a series of hand-designed features. Such a model could then be built up either through a successive cutting of the hypothesis space, or methods such as bayesian updating which track a probability distribution over models. Alternatively, the model representation itself could be learned in tandem with the rest of the system, although this would likely require large amounts of diverse and representative training data.

There are also a number of ways to get feedback with which to update our model once its representation is specified. One straightforward method is to make predictions about the user's behavior based on the current model and context, and treat the predictive power of the current model as a reward signal. In particular, the robot may make predictions about what kinds of actions will lead to the greatest probability that the user will satisfy the goals set out with the therapist, and receive a binary reward signal based on whether the user actually satisfies the goal. We assume that, all else being equal, the user would prefer to satisfy the goals they made with the therapist, which can give us additional information if some preference in the user unexpectedly prevents them from achieving these goals. We may additionally be able to use other information such as facial affect as auxiliary information with which to augment our update techniques, as in [21], though it is unlikely that such an approach would be sufficient by itself given current technical limitations in affect inference.

Capability inference is tightly linked with preference inference. This is because if a user has a hard time or simply cannot do an action, they will not do it, which is also what we would expect to see if they had the capability to do it but simply preferred not to. Purely from the perspective of RobOTA'a behavior, it will likely not matter much whether behavior is caused by preference or capability, as both cases cause similar predictions of user behavior and response to the robot's actions. However, this information may be important for other reasons, such as providing the therapist with advance warning of any important changes in a user's condition.

In many cases a preference against taking an action may indicate a problem in capability. For instance, if the robot infers that a user has a preference against lifting their left arm above shoulder height, that is suggestive that the user may be having problems with that shoulder. However, we cannot always be confident in such assumptions, and in general the only way to determine whether an observed (lack of) behavior is caused by preference or capability is to observe a circumstance in which the robot is confident that the user would perform an action if they were capable of it. The problem with this is that there are no guarantees we will naturally observe situations like this in the environment, and in fact such situations are likely rare. To address this the robot can attempt to explicitly bring such situations about, for example by asking the user to perform diagnostic motions while providing verbal indications of the importance of doing these motions.

Given all of this data, we still need a method of combining it into a coherent rating of the appropriateness of potential actions in the current context. There are many relevant approaches in the machine learning literature, the applicability of which depends on what kind of training data we have access to. For example, if we can gain access to "correct" labels for appropriateness ratings, which may be provided by human experts, we can use techniques in supervised learning. Alternatively, if all we have is access to environmental reward signals such as goal adherence or facial affect, we will need to rely upon techniques in reinforcement learning. Regardless of this choice, we will likely require a way to propagate the error or reward signals into the inputs of this

function, in order to train the other components in our system.

An important consideration for any adaptive system is how it will get the feedback that allows it to know how to tune its behavior. While auxiliary features such as facial affect may be relevant, such a robot should rely heavily on knowledge of whether its actions culminated in the successful completion of the therapist-specified goals. Thus, the robot must be able to translate between the goals as specified by the therapist and the kinds of observations that represent the successful completion of those goals.

This brings to light issues of robustness, such as the concern for wireheading if the robot can take actions which trick itself into observing a satisfied goal even when the goal was not actually satisfied. In addition, we need to ensure that the robot's specified goal-following behaviors properly respect the health and safety of the user in all relevant manners. It is important to avoid cases where the robot is incentivized to try and get the user to accomplish the therapist's goal at the expense of the user's own physical or psychological well-being. On the other hand, the robot cannot be so afraid of violating such rules that it refuses to take any action at all towards satisfying the therapist's goals. These are important open questions; for more discussion, see Section 2.8 on robustness and safety.

In the current work, we will focus our attention on building models of user preferences, for several reasons. First, preference models should be universally applicable and not particularly domain-specific, which does not hold as strongly for state or action models. This additionally means that such models may be useful outside of our focus on healthcare, and may be just as relevant for any assistant AI. As discussed above, a strong preference model can also serve as the baseline for capability models. We are now ready to specify in more detail the user-modelling framework we will be analyzing.

3.5 A Framework for User Modelling

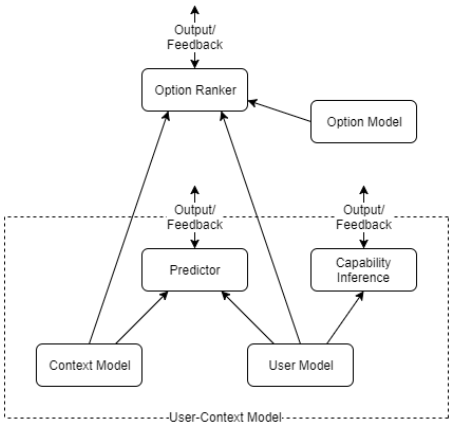
We will now introduce a zoomed-in picture of the user-context model, particularly of the user modelling, starting with the system's goals. We will then discuss proposals for how the preference inference procedure might be performed, and how it would be incorporated into the larger system.

In order to sufficiently satisfy the needs of adaptivity in the rest of the system, the user-model must satisfy some design criteria. First, the user model must be able to give usable features to the combined action-selection framework. How the combination of context, user, and option features is turned into option rankings is outside the focus on the current work, so our emphasis is simply on providing sufficiently expressive features that can be used as input to a unified model. Additionally, the user model must allow us to infer the capabilities of the user and how these change over time, as it must enable diagnostic data to be available to caregivers and medical experts. Finally, the model must be efficiently learnable over the course of interaction with the user.

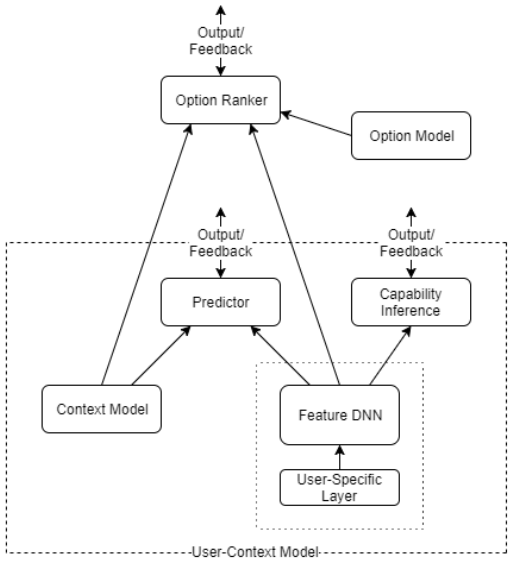
Figure 3.2a provides an architectural diagram of the user-context model. In this diagram, the option ranker and option model are the same as in figure 3.1. The context model transforms sensor data into relevant features usable both for option ranking and for making predictions (it encapsulates the environmental state). The user model is parameterized over the course of interaction with the user, and exports features usable for option ranking, user predictions, and capability inference.

In the language of reinforcement learning, the state would be a concatenation of the features for the context, the user, and the available options. Standard reinforcement learning techniques such as Q-learning and policy gradients could be used in the option ranking subsystem, with these features as input.

We can receive three different feedback signals in this way: a reward signal, a capability-



(a) A more detailed architectural diagram of the user-context model



(b) An alternative user-context model architecture where user features are extracted through a deep neural network

Figure 3.2: Proposed architectural diagrams for the user-context model

prediction signal, and a user-action-prediction signal. First, we receive a reward signal based on whether the user successfully completes the therapist-specified goals, which can propagate feedback through the option ranking subsystem and into all three inputs. Second, we can estimate the user's capabilities and use active learning techniques to query the user about their capabilities at some small cost, sussing out whether an observed trend is due to preference or capability limitations. The results of this query can also give our user model feedback in cases of misestimations. Finally, we can use techniques from inverse reinforcement learning to build up information about the user simply from observing their choices, which we can generalize as making predictions about their actions and updating when surprised.

As such, this model encapsulates the three design goals we had of the system, as the user model feeds features into the option ranker, is used for inferring capabilities, and has multiple paths for receiving evaluative feedback from the environment.

There are a few ways in which the user model could be structured to achieve the desired

effect. First of all, the model could consist of an expert-specified psychological model of human preferences. This approach would require the prediction and capability inference subsystems to also be designed in a manner that allows them to use this specific model. While there has been some work done on this front [16, 34, 35], more work needs to be done before such explicit models realistically capture a sufficient amount of human reasoning.

Alternatively, the features used to model the use could themselves be learned, for example as the internal representation in a neural network. Ideally, we would like the input to such a network to be a one-hot encoding of the user, as the identifier of a user does not itself encode features about the user. We can achieve this effect by having the input layer to the net be trainable independently for each user, while the higher levels are shared among all users and encode general preference-modelling information. This would require a large amount of training data over many different users in order to train the upper levels of the network to be sufficiently generalizable, but if such data is available then we will only need to learn a single layer's worth of weights during interaction with a new user, making subsequent training periods for new users much shorter. A diagram of this proposal is presented in figure 3.2b. As sufficient training data does not yet exist to train this model, we will focus on other methods, though in the future this approach would be a good direction for additional research.

Another approach is to model preferences in a manner similar to that used by Amin et al. [6]. This involves a task-independent reward signal that is added to the known task-specific reward signal. This task-independent reward function can be learned by observing the user act in a variety of known tasks, using a form of inverse reinforcement learning. This is the approach we will focus on, and it is further discussed and formalized in Chapter 4.

3.6 A Formal Specification of Preference Inference

The central question of this work is how to infer and model a user's preferences, such that an assistive robot can learn to personalize its behavior over time. Towards this end, we adopt the formalism of Amin et al. [6]. That is, a user is assumed to have an underlying task-independent reward function over states, θ, in addition to the standard task-dependent reward function R that encodes information about the specific short-term goal the user is attempting to reach. For example, when cleaning, R may encode information about higher rewards for states of higher overall cleanliness, while θ may encode information about concerns for safety or the avoidance breaking things, which we can expect to be present in all tasks that the user performs.

This formulation is useful because we can expect the user to act according to certain preferences no matter what task they are performing, and such preferences may be different for all users. In many cases we will have access to a kind of reward function for the task the human is doing, such as cleaning, but we do not expect all preference-related concerns to be encoded in that prior information. Thus, this formulation works well in the case where we know some aspect of the reward function for many of the tasks we can infer the human performing, and want to infer what additional preferences the user acts towards satisfying.

This is well-suited to our context, in which the robot has access to the therapist-specified goals related to rehabilitating the user. We assume that, all else being equal, the user will prefer to satisfy these goals if they are made aware of them. By this we mean that, if satisfying a goal was just as easy as not satisfying it, we assume the user will prefer to satisfy it. If this assumption does not hold, it will significantly complicate care, as getting the user to satisfy the goals involves getting them to act against their own preferences. While such cases may exist, our emphasis is on assisting the user, and users who do not want to be assisted fall outside the scope of the present work.

We now move on to a more complete specification of the robot's workflow and preference-

inference procedure. The robot begins the deployment phase with information about therapist-specified goals for the user, which it can prompt the user to complete in various task-specific ways. Once the robot has prompted the user to start the task, we assume the user will adopt the therapist-specified task-specific reward R as an additive component to their usual preference function θ, such that the true reward function Y is now equal to $R + \theta$. The robot will make predictions about how the user will behave based on its current model of θ, and observe the user acting in the environment. Based on the user's actual actions, in particular on whether the human successfully completed the goal, the robot will update its estimate of θ according to the newly-measured data. The robot can then use this improved model to make better predictions about when and how to prompt the user to complete tasks, as well as what interactive steps the user will find most useful in tasks with interactive components.

In addition to being relevant in the domain specified, there is reason to believe such a formalization will address some of the robustness and safety concerns laid out in Section 2.8. In particular, in this method the robot is uncertain of the true reward function Y, and assumes that the human knows this better than it itself does, similar to the cooperative inverse reinforcement learning paradigm proposed by Hadfield-Menell et al [40]. As they proved, this leads to proper incentives for the robot to avoid taking actions to manipulate or disempower the user, for example by finding a way to disable its own off-switch, while also incentivizing the robot to take actions it expects to be most helpful. However, this is not sufficient to address robustness, problems with which can manifest at any part of the system, including the reward function or state estimation. It is important to stress that all components of the system must be built with robustness and safety in mind in order for the system to perform reliably.

Chapter 4

Proposed Algorithms for Preference Inference

In this chapter we introduce a new preference-inference formulation: Observational Repeated Inverse Reinforcement Learning, or ORIRL, along with a proposed inference algorithm based on maximum-margin approaches. Some IRL approaches such as feature expectation matching attempt to infer enough about the user to predict their actions, without emphasis on learning the true underlying preference function. However, in many applications the true preference function itself may encode useful information. For instance, in rehabilitation applications, a clinician reward function may include relevant information about health and pain levels. In our case, we are trying to infer task-independent rewards to generalize the user's preferences to new tasks, which requires a model of the reward. Max-margin approaches are well-suited to our use-case, as they attempt to learn the underlying preference function directly [2].

The chapter is organized as follows. Section 4.1 introduces the Repeated Inverse Reinforcement Learning (RIRL) framework proposed by Amin et al. [6], which is the inspiration for ORIRL. This section includes a discussion on the assumptions made in RIRL, and why they are unsatisfactory for the settings in which RobOTA needs to operate. We then propose ORIRL

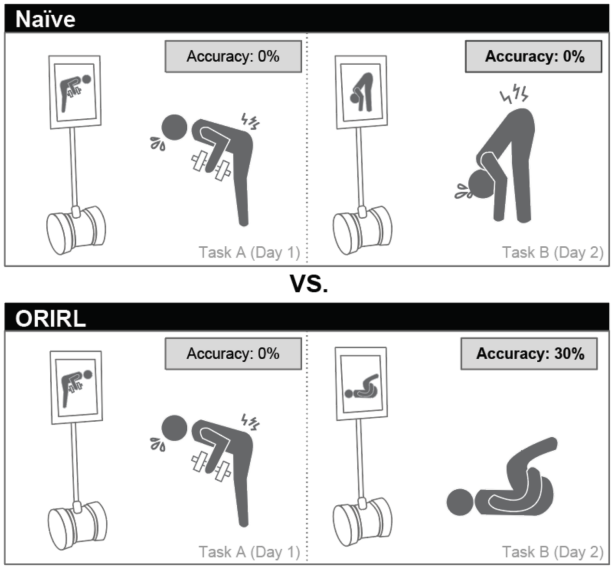


Figure 4.1: On Day 1, both ORIRL and a Naive algorithm achieves 0% accuracy on Task A. The next day, the Naive algorithm again achieves 0% accuracy on Task B, as it treats it as completely separate from the observed Task A. However, ORIRL is able to leverage what it learned about the user's underlying preferences on Day 1 to achieve a better accuracy on Day 2.

in Section 4.2 as a variant with less restrictive assumptions. Finally, Section 4.3 describes a max-margin learning approach that solves ORIRL's problem statement by inferring a global reward function purely from observation.

4.1 RIRL Formulation

RIRL and ORIRL model tasks and environments in terms of Markov decision processes (MDPs). An MDP is defined by the state space S, action space A, transition dynamics $P : S \times A \to \Delta(S)$, initial state distribution $\psi \in \Delta(S)$, discount factor $\gamma \in [0,1)$, and reward $Y : S \to \mathbb{R}$. In addition, an agent's strategy in an MDP is denoted by their policy $\pi : S \to \Delta(A)$, which determines the (possibly stochastic) action to take in any particular state. The state inhabited at timestep t is denoted as s_t.

In many cases of interest, a full Markovian state-space formulation for the MDP is necessarily large or infinite, leading to problems with tabular reinforcement-learning methods

[91]. This is commonly addressed through a mapping ϕ from states to low-dimensional state features. This is extended to a function μ that maps policies to the expected exponentially-discounted sum of state features under that policy, known as feature expectations [1]. That is, $\mu(\pi) = \mathbb{E}_\pi \sum_i \gamma^i \phi(s_i)$. The reward function is commonly assumed to be approximately linear in these state features, so that the reward is parameterized by a vector ω. Thus, the expected discounted reward from a policy π is just the dot product between ω and $\mu(\pi)$.

In RIRL, the true reward for a given task is a weighted sum between a task-independent reward term and a task-dependent reward term. With a linear approximation ω of the reward function, this means for task i that $\omega_i = \theta + R_i$, where θ is the task-independent reward term and R_i is the task-dependent reward for task i. The task-dependent reward terms R_i are assumed to be known fully ahead of time, and the goal in RIRL is to infer the task-independent reward θ through interaction. This lets the robot adapt to new tasks much more quickly, as it can understand ahead of time how the user's preferences will interact with the task's structure.

The interactive setting of RIRL itself relies on several assumptions. In this setting, a robot attempts to perform a task on behalf of a user. The robot and user both know the complete task-dependent reward function, but the user additionally has some set of global preferences θ that they are trying to satisfy. The user carefully observes the robot's performance, and identifies whenever the robot selects an action with expected loss of more than ε compared to the optimal action. Then the user replicates the state in which the robot made the mistake, and demonstrates an optimal trajectory from this state. With this information, the robot can update its model in such a way as to guarantee with high probability that the number of mistakes it will make is bounded by an amount that depends on ε, the size of the state space, and the desired probability.

4.2 ORIRL

The assumptions made for RIRL are unsatisfactory in the context of RobOTA for two reasons. First, we are interested in many settings in which the interactive context of RIRL is infeasible. Many users do not have the technical expertise to identify ε-suboptimal actions, enter a context that exactly matches the robot's internal state representation when it made that action, and demonstrate an optimal trajectory. In addition, this interactivity implicitly assumes that the user and robot have identical capabilities. If the robot's morphology is different than the user's, then the robot may not be able to perform the actions the user is expecting, and the user may not be able to tell whether an action is optimal with respect to the robot's form. This is especially problematic in the context of OT, in which performance in many activities relies very heavily on the exact biomechanics of the user's body.

Secondly, RIRL assumes the robot always has perfect prior knowledge of the task-dependent reward R_i for every task in which it engages with the user. It is reasonable to assume prior information of the task, as many of these tasks are predefined by OT experts, but assuming perfect information of task-dependent rewards also implicitly assumes knowledge of how motivated the user is by the completion of various tasks. If a user is more or less motivated by a task, they may be more or less willing to trade off progress towards that task against their overall preferences. A robot that assumes perfect knowledge of task rewards will be unable to adapt to users with differing levels of motivation for different tasks.

To address these drawbacks, we introduce a variant of RIRL known as Observational Repeated Inverse Reinforcement Learning (ORIRL). ORIRL does not assume the user is willing and able to critique the robot's performance and demonstrate optimal trajectories, nor that the robot has a sufficiently similar morphology to enable one-to-one mappings of actions. The "Observational" part of ORIRL indicates that an ORIRL agent should infer the user's task-indpenedent preferences *solely from observation*, without any assumption that the user will be

able to aid in the inference.

In addition, ORIRL agents only have *partial* information of the task-dependent reward terms R_i; namely they do not know the relative magnitudes of the terms. This means that $\omega_i = \theta + k_i R_i$, where k_i is the weighting for task i. We define K as the vector of scalars $K = [k_1, k_2, \ldots, k_n]$, where n is the number of tasks. Thus the goal of an ORIRL agent is to infer θ and K from observation.

ORIRL better captures the kinds of inference needed to make RobOTA adaptive, but it is strictly more challenging than RIRL as it involves inferring more properties from less data. The original algorithm presented by Amin et al. [6] to solve RIRL doesn't extend to ORIRL, as it requires the user to identify mistakes and provide optimal alternative trajectories. Despite these challenges, we aim to establish that inference is still feasible in the ORIRL setting. Section 4.3 presents an algorithm to perform this inference, and Chapter 5 presents empirical validation of the algorithm's performance.

4.3 Max-Margin ORIRL

The original IRL problem as defined by Abbeel and Ng [1] was to infer a reward function that matched the observed behavior, assuming the expert behaves optimally. However, this leads to problems of ambiguity, as there are many reward functions that would explain any given observation. The problem is exacerbated if the expert's demonstration are suboptimal, as will be the case if the expert is a human acting in a complex environment, such as on a busy road or in an operating room.

In order to overcome the challenges inherent in the original IRL formulation, Ratliff et al. [78] proposed the use of max-margin methods. In this formulation, they first turn the feasibility problem into one of optimization – namely they minimize the L2 norm of the weight vector, as a form of complexity penalty, subject to the optimality constraints. This means there will be

a unique solution, but since they are minimizing the L2 norm, the reward function that always returns 0 will be selected for any given set of observations, despite the fact that humans rarely have exactly no preference over any possible state.

To deal with this class of problem, maximum-margin methods incorporate the heuristic that the expert's policy is likely significantly better than alternatives. Structured-prediction maximum-margin methods require the expert's observed policy to not only match, but beat all other policies by an amount that scales with a measure of difference between the policies. This encodes the idea that "nearby" policies may be nearly as good as the expert's policy, but "faraway" policies are probably worse than the optimal policy by a larger amount.

This still leaves the unsatisfactory assumption that the expert's policy is exactly optimal with respect to the hidden reward. In addition, while it may be a good heuristic that the expert's policy is significantly better than others, this may not always be the case. Maximum-margin methods handle these concerns by including the "slack" variable, ξ, that can allow policies to be close to or better than the expert's policy, at some cost in the optimization term. Thus, the full optimization problem becomes one of solving:

$$\min_{\omega,\xi} ||\omega||_2^2 + C\xi \tag{4.1}$$

$$\text{s.t.} \qquad \omega^\top \mu(\pi^*) \geq \omega^\top \mu(\pi_i) + m(\pi^*, \pi) - \xi \qquad \forall \pi \tag{4.2}$$

Where π is the expert's policy, π^* is the optimal policy, ω represents the weights for the reward function, and $m(.)$ is a distance function which compares the optimal policy π^* and the expert's policy π.

We can additionally extend this to multiple MDPs that share the same reward function by using:

$$\min_{\omega,\xi_i} ||\omega||_2^2 + C\sum_i \xi_i \tag{4.3}$$

$$\text{s.t.} \qquad \omega^\top \mu(\pi_i^*) \geq \omega^\top \mu(\pi_i) + m(\pi_i^*, \pi_i) - \xi_i \qquad \forall i, \pi_i \tag{4.4}$$

The optimization term is quadratic and the constraints are linear, allowing efficient solutions for a given number of constraints using quadratic programming. However, there is a constraint for every possible policy, which may be large or infinite. Ratliff et al. [78] address this by modifying the form of the constraints and using subgradient methods, but a simpler alternative is iterative constraint generation of the form used by Abbeel et al. [1].

We now modify the existing maximum-margin algorithm described above to extend it to the context of ORIRL. In ORIRL, $\omega_i = \theta^* + k_i R_i$, where R_i is the task-dependent reward vector that is assumed to be known in advance, up to scaling. The goal is to infer θ^* (the user-specific task-independent reward term) and $K = [k_1, k_2, \ldots, k_n]$ (the proper scaling terms for the task rewards). Thus we expand ω in the constraint terms. To get the analogue of a complexity constraint in the minimization term, we minimize the L2-norm of θ^*, our inferred task-independent reward vector. In order to encode a similar regularization over the values of K_i we must also include a penalty for the L2 norm of $K - \hat{1}$, where $\hat{1}$ is the constant vector of all 1's, to regularize the values of K to be closer to 1.

We are left with the following equation:

$$\min_{\theta^*, K, \xi_i} ||\theta^*||_2^2 + B||K - \hat{1}||_2^2 + C\sum_i \xi_i \tag{4.5}$$

$$\text{s.t.} \qquad (\theta^* + k_i R_i)^\top \mu(\pi_i^*) \geq (\theta^* + k_i R_i)^\top \mu(\pi_i) + m(\pi_i^*, \pi_i) - \xi_i \qquad \forall i, \pi_i \tag{4.6}$$

With this formulation, the optimization remains quadratic and the constraints remain linear, allowing efficient solution through quadratic programming methods.

Chapters 4 and 5, in part, have been published as it appears in "Preference Learning in

Assisitive Robotics: Observational Repeated Inverse Reinforcement Learning", Machine Learning for Healthcare Conference 2018 [102]. The thesis author was the primary investigator and author of this paper.

Chapter 5

Experimental Validation and Results

In Chapter 4 we introduced a max-margin algorithm for performing inference in the ORIRL setting, but it still must be seen whether this proposal performs adequately in a realistic setting. This chapter introduces a realistic OT setting that RobOTA should be able to operate in, and discusses the performance of max-margin ORIRL in inferring user preferences. Section 5.1 discusses the context of the experiment, and why an experiment of this kind was chosen as a testbed. Then we discuss the details of the experiment itself in Section 5.2. The results are presented in Section 5.3, and their significance is discussed in Section 5.4.

The work in this chapter was done in collaboration with Francesco Ferrari, Teofilo E. Zosa, and Professor Laurel D. Riek. Professor Riek and Teofilo Zosa helped design the experimental setting itself. The robotic interaction was coded by Teofilo Zosa, who also oversaw the experiment and collected the raw data. Programming for the max-margin ORIRL implementation was aided by Francesco Ferrari and Teofilo Zosa. Francesco Ferrari helped analyze the results and compiled them into a published paper [102], which this chapter is built upon.

5.1 Experimental Context

The motivating context for this work is in healthcare. In particular, we are interested in methods for inferring preferences from users who have been given non-binding therapeutic advice from clinicians, and who must then figure out how to balance engaging in that activity given their own preferences. For example, users might be advised to perform stretching exercises twice a day.

This is a particularly interesting application space for two main reasons. First, this is a common scenario in ambulatory care – clinicians provide prescriptive advice which may or may not be followed by users, or may be only followed for a short time, etc. (e.g. adherence issues). However, if we can infer their preferences in these scenarios and build interactive, adaptive systems based on them, it can have a substantial practical impact – tailored, individualized treatment plans are far more likely to be adhered to [60, 42, 66]. This is a key application area for RobOTA.

Second, the clinician's advice can influence the choices that the user might make, and hence the reward of corresponding tasks. Because of this, our system will not only learn the user's preferences, but also the expert's advice. Thus, we evaluate our methods in a rehabilitation setting across multiple activities, in which participants receive written advice from a physical therapist, and then interact with a robot facilitator to choose a set of activities to perform.

In our study, participants participated in a week-long, twice-a-day prehabilitation activity session with a robot (See Section 5.2 and Figure 5.2). The activity sessions lasted for 10 minutes, wherein the robot solely provided instruction and structure. In each session, participants would be greeted by the robot and presented with advice from a licensed physical therapist (randomized by day and participant). This clinical advice was related to the different categories of activities to bias activity selection. The activity categories consisted of eight different parts with standing or not standing versions for a total of 16 unique prehabilitative activities.

Participants would then navigate to a main activity screen where they could choose the activity they would like to perform. They would then be presented with an instruction screen which contained a video example of the activity and brief explanatory text. Once ready, the participants would then begin the activity, at which point a session timer would start. Participants were allowed to perform an activity for as long as they liked.

Once participants were finished, they would then navigate back to the main activity selection screen which would pause their session timer. Two robot-initiated events could occur at this point: following the first activity that brought session time over 5 minutes, the robot would prompt the user for a break. If total session time exceeded 10 minutes, the robot would end the session.

5.2 Participants and Procedure

We recruited four participants to participate in our study. Their ages ranged from 20-24 (average age = 22 years). Three participants were women, and 1 was male. They were primarily undergraduate and graduate students who spend a majority of their time sitting, working at computers. No participants reported any pre-existing health conditions which would affect study participation.

5.2.1 Exercise Choices

Based on feedback from a licensed physical therapist, we focused on activities that would reduce susceptibility to repetitive stress injury (RSI) in office workers. The activities in our study focused on eight body parts, with a standing version and a non-standing version (either seated or lying down) for each, yielding a total of 16 exercises. Table 5.1 lists the six major pathologies we focused on improving, and their corresponding eight body parts.

Forward head posture results from improper placement of computer monitors, as well as

Table 5.1: Prehabilitative activities for RSI prevention.

Pathology	Forward Head Posture	Kyphotic Posture	Wrist Tension & Irritation	Pelvic Tilt	Prolonged Hip & Knee Flexion	Limited Hallux Dorsiflexion
Body Part	Neck	Pecs & Traps	Wrists	Back	Hips & Hamstrings	Toes
Non-Standing Variation	Lying	Lying	Lying	Lying	Lying (Hips) & Sitting (Hamstrings)	Sitting

the stationary nature of these devices. This causes flexion of cervical spine with extension at the occiput-C1 joint which leads to overstretched/lengthened cervical neck flexors (especially deep neck flexors) and tight/short upper cervical/occiput extensors. Kyphotic posture (i.e. rounded shoulders) arises from excess forward leaning common in prolonged computer work. This commonly leads to tight/short anterior chest muscles (e.g. pecs), tight/short upper trapezius and levator scapulae, overstretched/lengthened posterior shoulder/upper back muscles (e.g. external rotators of the shoulder, middle/lower traps), and kyphotic (flexed) thoracic spine. Wrist tension and irritation, specifically in the wrist extensors and flexors, occurs as a consequence of maintaining neutral grip and wrist position when having to use a keyboard and mouse. This also leads to overuse of finger flexors.

Pelvic tilt, both anterior and posterior in the sagittal plane, occur as a consequence of prolonged periods of sitting. For example, shorter people may tend to sit in more posterior pelvic tilt in order for their feet to touch the floor ("slouched" pelvic positioning). Taller people either sit in anterior pelvic tilt (they have no problem getting their feet to touch the floor) or posterior pelvic tilt (poor sitting posture, slouched or reclined positioning). People who sit in a very upright and "proper" position (90/90 position) may have lower back pain (overuse of back extensors) as well as overactive hip flexors to maintain this position. Prolonged hip and knee flexion also occurs as a result of sitting for extended periods of time. This leads to tight/short hamstrings,

sway back posture while standing, and often accompanies posterior pelvic tilt. Limited hallux (i.e. big toe) metatarsophalangeal dorsiflexion and extension also often occurs in office workers and is typically the result of the use of certain types of shoes (such as heels), midfoot pronation, being overweight or sedentary, and frequent running. This can lead to conditions such as plantar fasciitis. Very limited dorsiflexion and extension (i.e. less than 60°) adversely affects normal gait. All pathologies and etiological explanations were provided by the licensed physical therapist in our study [108].

In addition, to emulate expert advice given for multiple tasks, there were six sets of advice provided by a therapy expert, targeting the six different pathologies. Each set of advice refers to a known expert-specified reward function over state features. We emphasize simple, easy to specify reward functions that give a positive reward when the task is completed and no reward everywhere else. For instance, a set of advice targeting neck posture would have a positive reward only when designated neck stretches and exercises are completed. Because we want to learn the preferences of the users, we do not assume to know ahead of time to what extent the users are influenced by the therapist's advice, and users are encouraged to perform whichever activities they want to perform regardless of the advice.

5.2.2 Equipment and Interaction Design

In our study, we utilized a Double robot, and two small exercise mats (See Fig. 5.2). The robot ran a custom application designed specifically for the study. In accordance with best practices in the health behavior change community [42, 71, 104], all messaging in the application was delivered in an positive manner, e.g., "Welcome [User]! It's so great to see you again!" and "You did great, I can't wait to see you again!". Furthermore, illustrated smiling faces were used at the introduction, break, and outroduction screen, respectively.

A robot's morphological and behavioral features has been shown to have a significant impact on attentional capture and user engagement [58], which can affect both the quality of data

gathered and the algorithmic performance that depends on that data. For this reason, the robot's height was dynamically adjusted at certain parts of the study to promote greater engagement. At the start of the study, the robot would be initially set to its minimum height (about three feet from the ground) to simulate that the robot was asleep/off. Once the user began a session, the robot would raise its display to its maximum height, which corresponded to eye level for our participants (about five feet from the ground). When choosing non-standing activities, the robot would lower itself back to its minimum height to help maintain a comfortable viewing angle. Once the participant completed this activity, the robot would then raise itself back to its maximum height. To ensure a comfortable interaction, the users had the option of adjusting the height manually at any point in the session.

Finally, to simulate cooperative behavior, the robot would provide the participant with an encouraging remark when the user completed an activity after the five minute mark of the study. The robot would then inform the participant that it was itself motivated to take a stretching break. It would redisplay its smiling face, and turn 360 before prompting the user that it was ready to continue. This was the only time the robot moved, and its kickstands were deployed immediately after to ensure user safety.

5.2.3 Procedure

At the beginning of the study, participants participated in an orientation session. An experimenter taught them how to properly interface with the robot and performed demonstrations of the 16 different prehabilitation exercises of the study. Participants participated in approximately two sessions per day, five times per week (Monday - Friday), for one week.

Prehabilitation sessions consisted of participants entering the experimental area, where the interactive robotic facilitator (see Fig. 5.1) would take them through the day's activities, as explained in Section 5.1. To mitigate possible ordering effects, exercises presented on the robot's display at the activity selection screen was randomly chosen for each session. Additionally, to

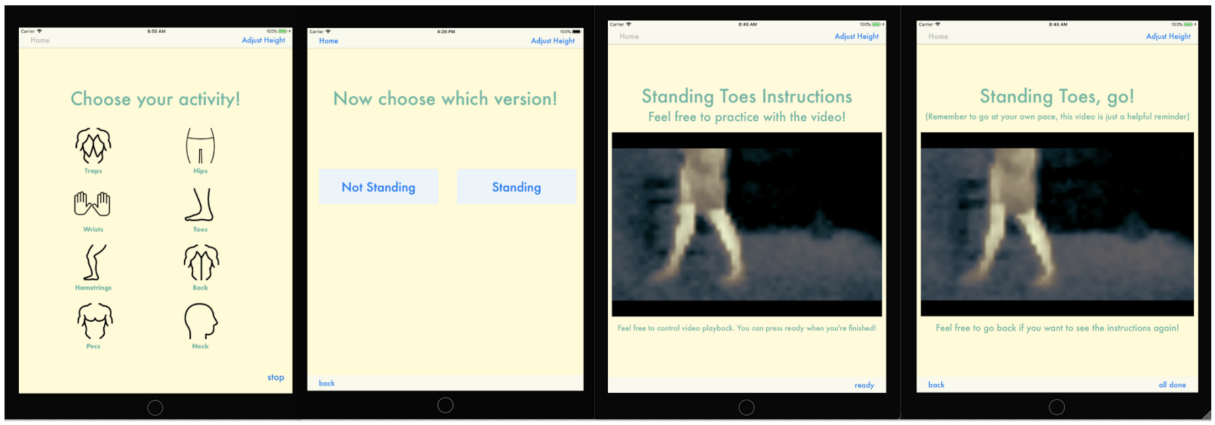


Figure 5.1: (a): The main activity selection menu. (b): The standing/not standing version selection screen. (c): A sample instructions screen (d): A sample of the screen that plays while the user performs an activity.

minimize gender and/or cultural bias, all videos and pictures were presented as thermal images or silhouettes.

5.2.4 Data Collection and Feature Engineering

In our experiment we used the following measures: the group of the selected action, its duration, and whether the user performed the seated or standing variation of the activity. The action's group is linked to a unique identifier and it tells us which muscle group was activated during the exercise. The action's duration was measured in seconds and was also used to calculate the total duration of a session. Finally, the action's sit/stand variation was recorded as a boolean variable.

During each activity session, each participant read the session's prompt and chose a sequence of actions to perform in that session. To emulate a home care setting, no human facilitators or other participants were present during the sessions; each participant's activities were classified by the interactive visual dialogue system on the robot's touch screen. Each session generated data consisting of the participant's ID, the session ID, and the sequence of actions the participant selected, which includes the three measures described above. Each action is then

Figure 5.2: Left: the robot in a real-world setting. Right: A user interacting with the robot.

encoded as a vector that combines a one-hot encoding of its targeted muscle group, its discretized duration, and whether it was the sitting or standing variation. In addition, the session is linked to the advice given in that session.

This dataset is provided to ORIRL as a set of feature expectations generated by the participant's actions. That is, the algorithm models the set of sessions as a Markov Decision Process (MDP), and observes the empirical discounted sum of state features resulting from the participant's choices.

The features provided to the algorithm approximate the relevant information about the actions and state. In addition to the action features described above, the state was mapped by concatenating the following vectors: $(V_i, V_i^2, V_i\dot{V}_{i-1}, X_i, X_i^2, X_i\dot{X}_{i-1}, T_i$. V_i corresponds to an array that keeps count of the variations (standing or sitting) up to state i, X_i corresponds to an array that counts the muscles targeted up to state i, and T_i corresponds to a one-hot encoding vector which discretizes the total duration of the session up to state i. These features were selected to give ORIRL sufficiently expressive representations with which to model preferences, without biasing the algorithm by providing extra prior data. For instance, the algorithm initially has no prior expectation that participants might tend to prefer to take varied actions in each session, and

must learn such associations from observation.

5.3 Results

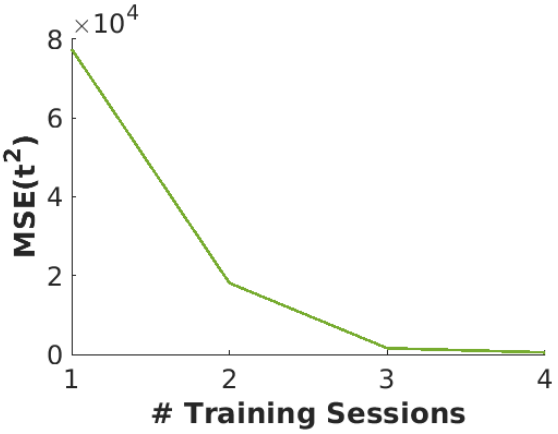


Figure 5.3: The mean squared error between max-margin ORIRL's prediction of the next session's duration and the empirical duration of that session, plotted against the number of sessions the system has already observed for this user. The graph shows how with more sessions ORIRL is able to make better predictions on the time durations of new tasks.

Ground truth data for user preference models is infeasible to obtain, so we measure the performance of the algorithm by its predictive power and its convergence properties.

To see how the predictive power of the model changes as it is given more sessions to train on per user, we gave the model the first n sessions for each user and measure how well it predicts aspects about session $n+1$. Namely, we measured the mean squared error between the time features of the user's actions with the time features of the policy that is optimal with respect to the inferred preferences. Unlike the other features, time is continuous, which allows a more straightforward error analysis. The results in Figure 5.3 show how error decreases as more sessions are provided to train on, with diminishing returns.

To track the convergence rate of the algorithm, we run the algorithm until it converges to an estimate θ of the user's task-independent preferences. We then observe the mean squared error of the estimate θ_i after iteration i with respect to the final converged θ. Figure 5.4 plots this, in

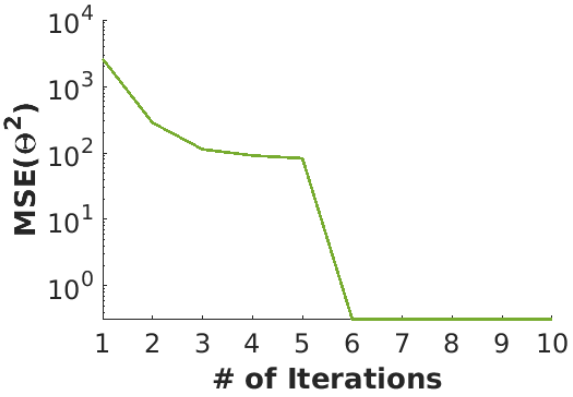


Figure 5.4: MSE of θ at each iteration of ORIRL, relative to the final inferred value. The algorithm learns the task independent features by iteratively applying max-margin on the inferred reward functions.

which we observe a rapid convergence in the first few iterations, until after iteration 6 the weight values for the inferred task-independent preferences only undergo small adjustments.

Finally, in order to get a more general estimate of predictive power beyond just predicting the time variable, we define an error metric based on the negative log-likelihood of observing the user data under the inferred preferences. In the vein of Ramachandran and Amir [77], we assume a generative model where a user selects actions based on an exponential distribution over the Q-values of that state-action pair. That is: $P(a|s,R) = \frac{1}{Z}e^{\alpha Q^*_R(s,a)}$, where α is a hyperparameter weighting how "noisy" the user's choices are, $Q^*_R(s,a)$ is the Q function that gives the expected discounted sum of rewards from following the optimal policy after taking action a in state s, and Z is a weighting term to make the probabilities for all actions sum to 1.

Given this generative model, we can compute the error score as the negative log-likelihood of observing the user data under the total reward function inferred by the ORIRL model. That is, for each estimated total reward, we define a probability function over the set of actions selected in the observed states, and rate the model based on how much probability mass its inferred reward function assigned to the set of true user actions. Figure 5.5 plots this score per iteration for each task the model is inferring. The first iteration bases its predictions off a random policy, and we

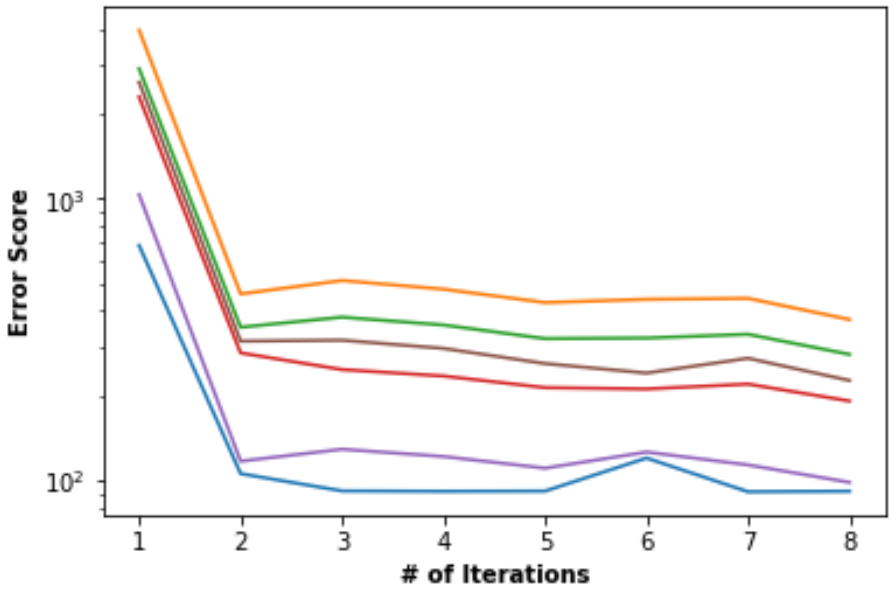


Figure 5.5: Averaged prediction error scores per task. Prediction error scores are based on the negative log-likelihood of observing the user-generated dataset under the inferred preferences at each iteration.

observe a very significant improvement in predictive power after just one additional iteration. Afterwards, the error score continues to steadily improve (note the logarithmic scale on the Y-axis means improvement slows over time).

5.4 Discussion and Future Directions

Overall, the results we obtained with max-margin suggest that it is possible to learn task-independent preferences in this framework (See Figs. 5.3 and 5.5) in a small number of iterations (See Fig. 5.4).

The algorithm achieves a sensible time inference accuracy on unseen sessions (see Figure 5.3), suggesting that the inferred user preference model is able to successfully capture relevant information about the user's task-independent preferences. Moreover, max-margin ORIRL is able to successfully learn the global preferences of a user across different real-world tasks (see Figure 5.5). For example, ORIRL learned that one user prefers to stand when performing an exercise,

and also prefers activities involving the back. It can then transfer these general inferences to better predict the user's behavior even on tasks it has never seen the user perform before.

Inferring user preferences in observational settings, as ORIRL does, will greatly benefit robotics by promoting more personalized long term interactions between robots and humans. A robot with a stronger understanding of the user will be able to better handle uncertainty introduced by new situations. For example, they can rely upon previous knowledge in new situations, as was demonstrated when the algorithm predicted features of unseen tasks (See Figure 5.3).

While ORIRL's observational nature means it relies on weaker assumptions than previous work, it does assume that the robot can infer which task the user is performing. This turns out to be a sensible assumption, as it can often be done for distinctive tasks [90, 8]. However, even in cases where there is not an easy way to identify tasks *a priori*, Babeş-Vroman et al. [12] demonstrate that clustering approaches can be used to group demonstrations together based on the task that generated them.

Chapters 4 and 5, in part, have been published as it appears in "Preference Learning in Assisitive Robotics: Observational Repeated Inverse Reinforcement Learning", Machine Learning for Healthcare Conference 2018 [102]. The thesis author was the primary investigator and author of this paper.

Chapter 6

Conclusion

6.1 Contributions

The application of robotics to healthcare has vast untapped potential. Robots could support caregivers, assist health care workers, and provide consistent, personalized care to patients. This thesis explores the use of assistive robots to support personalized rehabilitation in the home. We propose occupational therapy as a natural testbed, and examine some relevant trends in related technology, such as workflow detection and reinforcement learning. These paint a concrete picture of where more work is needed to enable the kinds of robotic assistants that the field envisions. In particular, it is vital to ensure that such robots are deeply adaptable to the personal needs and preferences of their users.

To make assistive robotic applications more concrete, we proposed a specific set of goals as a target, including flexible specification, seamless adaptivity, and low overhead of adoption. To further support these goals, we propose a high-level system architecture for exploring this research area in Chapter 3. By doing this, we decomposed the problem and generated tractable research directions. To demonstrate the usefulness of this decomposition and to begin the process of building these systems, we narrowed in on the problem of providing adaptive behavior through

modeling the user's preferences.

After building a formal definition of the real-world preference inference problem of interest, we introduced, developed, and analyzed a novel paradigm: Observational Repeated Inverse Reinforcement Learning (ORIRL). In ORIRL, robots must operate in realistic, limited-information, and limited-interaction settings. After analyzing how existing methods fail to extend to this domain, we introduced a maximum-margin algorithm for performing the ORIRL inference. We validated that maximum-margin ORIRL successfully and efficiently learns users' preferences through an experiment with real users in an occupational-therapy domain.

6.2 Future Work and Open Questions

In the future, we seek to improve the features ORIRL uses to map each action taken by the user. We could improve the current feature set by learning the features themselves, as in deep neural networks. This would also have the benefit of making ORIRL easier to deploy without requiring hand-crafted features, at the cost of requiring much more data than we had the resources to collect. One possible way to tackle the lack of data is to apply a deep generative model on a pre-existing smaller dataset to generate new user data using a different Gaussian distribution. Another option is to craft stronger priors, so that the algorithm only has to learn how each user differs from some "average", rather than learning preferences from scratch.

In addition, we plan to explore new applications of ORIRL. This work could be extended to content filtering in more interactive recommender systems, such as a system that recommends nearby social events to help engage socially isolated older adults (a known public health problem [70, 87]). Instead of modeling a user's social preferences as a static mapping, the system could have a more dynamic, interactive model of the user's needs and preferences. That is, learning preferences as task-independent rewards like RIRL and ORIRL do would enable them to better take context into account, such as the history of recent social interactions and preferences for

novelty or stability. This would enable different recommendations that adapt to the user's actions and environment, including interactions with the recommender itself.

The preference-learning done by ORIRL aims to satisfy the goal of building a user-context model as described in Section 3.2.1. Future work will target the goal of incorporating preference-learning into a comprehensive model that can rank the appropriateness of actions. Such a model will need to incorporate not only preferences, but also an understanding of how those preferences interact with the particular context in which the user and robot are currently acting. How to best model context in the domain of occupational therapy and how to design the interaction between context and preferences are currently open problems.

Finally, the user-context model is just one component of the RobOTA architecture described in Section 3.2, and will need to be incorporated into the larger system. How to define the interaction between the execution graph and the user-context model is an open problem in this integration. Another open question is how we can define an interface to allow for occupational therapists to easily and expressively define goals for the system to pursue. Section 3.2.2 discussed open questions relating to the other components in this architecture in more detail.

6.3 Closing Remarks

The goal of this work is to enable adaptive healthcare robotics systems. With a rising need for healthcare that is outpacing the available resources, building scalable robotic solutions to fill this gap is more important than ever. Such systems could aid human healthcare experts by implementing expert-defined intervention strategies with users. Ideally, robots like these could be deployed long-term in a user's home, enabling more graceful, independent care and aging.

Towards this goal, the current work provides clarity on the necessary existing research effort needed to bring about such a system. The ORIRL framework described in this work provides both a useful standalone tool, as well as an example of using this roadmap to generate

and solve tractable research questions. By itself, the ability to model a user's preferences across different situations over a long period of time will enable more personalization and improved collaboration between the user and a robot. This is especially important now as robots are becoming more involved in our daily routines, and commonly operate inside our homes.

We envision a future in which critical and currently-unmet needs can be satisfied by experts leveraging robotic assistants. It is our hope that future work continues to expand these ideas and ensure that this idea comes to fruition with robots that are truly personalized.

Bibliography

[1] ABBEEL, P., AND NG, A. Y. Apprenticeship learning via inverse reinforcement learning. *Proceedings of the 21st International Conference on Machine Learning (ICML)* (2004), 1–8.

[2] ABBEEL, P., AND NG, A. Y. Inverse reinforcement learning. In *Encyclopedia of machine learning*. Springer, 2011, pp. 554–558.

[3] AGGARWAL, J. K., AND RYOO, M. S. Human activity analysis: A review. *ACM Computing Surveys (CSUR) 43*, 3 (2011), 16.

[4] AKGUN, B., CAKMAK, M., YOO, J., AND THOMAZ, A. Trajectories and Keyframes for Kinesthetic Teaching: A Human-Robot Interaction Perspective. *ACM/IEEE International Conference on Human-Robot Interaction* (2012), 391—-398.

[5] AL-AZAWEI, A., AND BADII, A. State of the art of learning styles-based adaptive educational hypermedia systems (ls-baehss). *International Journal of Computer Science & Information Technology 6*, 3 (2014), 1–19.

[6] AMIN, K., JIANG, N., AND SINGH, S. Repeated inverse reinforcement learning. In *Advances in Neural Information Processing Systems* (2017), pp. 1815–1824.

[7] AMODEI, D., OLAH, C., STEINHARDT, J., CHRISTIANO, P., SCHULMAN, J., AND MANÉ, D. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565* (2016).

[8] ARBAB-ZAVAR, B., CARTER, J. N., AND NIXON, M. S. On hierarchical modelling of motion for workflow analysis from overhead view. *Machine Vision and Applications 25*, 2 (2014), 345–359.

[9] ARGALL, B. D., CHERNOVA, S., VELOSO, M., AND BROWNING, B. A survey of robot learning from demonstration. *Robotics and autonomous systems 57*, 5 (2009), 469–483.

[10] ARON, J. How innovative is apple's new voice assistant, siri?, 2011.

[11] AYRES, R., AND MILLER, S. Industrial robots on the line. *The Journal of Epsilon Pi Tau 8*, 2 (1982), 2–10.

[12] BABES, M., MARIVATE, V., SUBRAMANIAN, K., AND LITTMAN, M. L. Apprenticeship learning about multiple intentions. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)* (2011), pp. 897–904.

[13] BASS, J. D., AND BAKER, N. A. Occupational therapy and public health: Advancing research to improve population health and health equity, 2017.

[14] BEHZADAN, V., AND MUNIR, A. Vulnerability of deep reinforcement learning to policy induction attacks. In *International Conference on Machine Learning and Data Mining in Pattern Recognition* (2017), Springer, pp. 262–275.

[15] BENSON-TILSEN, T., AND SOARES, N. Formalizing convergent instrumental goals. In *AAAI Workshop: AI, Ethics, and Society* (2016).

[16] BERGEN, L., EVANS, O., AND TENENBAUM, J. Learning structured preferences. In *Proceedings of the Annual Meeting of the Cognitive Science Society* (2010), vol. 32.

[17] BERNARD, J., CHANG, T.-W., POPESCU, E., AND GRAF, S. Using artificial neural networks to identify learning styles. In *International Conference on Artificial Intelligence in Education* (2015), Springer, pp. 541–544.

[18] BEYGELZIMER, A., DASGUPTA, S., AND LANGFORD, J. Importance weighted active learning. *arXiv preprint arXiv:0812.4952* (2008).

[19] BOESL, D. B., AND LIEPERT, B. 4 robotic revolutions-proposing a holistic phase model describing future disruptions in the evolution of robotics and automation and the rise of a new generation rof robotic natives. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on* (2016), IEEE, pp. 1262–1267.

[20] BROCHU, E. *Interactive Bayesian optimization: learning user preferences for graphics and animation*. PhD thesis, University of British Columbia, 2010.

[21] BROEKENS, J. Emotion and reinforcement: affective facial expressions facilitate robot learning. In *Artifical intelligence for human computing*. Springer, 2007, pp. 113–132.

[22] BRUSILOVSKY, P., AND MILLÁN, E. User models for adaptive hypermedia and adaptive educational systems. In *The adaptive web*. Springer, 2007, pp. 3–53.

[23] BUSS, D. M. Social adaptation and five major factors of personality.

[24] CADENE, R., ROBERT, T., THOME, N., AND CORD, M. M2cai workflow challenge: convolutional neural networks with time smoothing and hidden markov model for video frames classification. *arXiv preprint arXiv:1610.05541* (2016).

[25] CHRISTIANO, P. F., LEIKE, J., BROWN, T., MARTIC, M., LEGG, S., AND AMODEI, D. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems* (2017), pp. 4299–4307.

[26] CLARK, J., AND AMODEI, D. Faulty reward functions in the wild. https://blog.openai.com/faulty-reward-functions/. Accessed: 2018-12-23.

[27] CLEVER, D., AND MOMBAUR, K. D. An inverse optimal control approach for the transfer of human walking motions in constrained environment to humanoid robots. In *Robotics: Science and systems* (2016).

[28] COHN, D. A., GHAHRAMANI, Z., AND JORDAN, M. I. Active learning with statistical models. *Journal of artificial intelligence research 4* (1996), 129–145.

[29] CUSTERS, R., AND AARTS, H. Positive affect as implicit motivator: on the nonconscious operation of behavioral goals. *Journal of personality and social psychology 89*, 2 (2005), 129.

[30] DORÇA, F. A., LIMA, L. V., FERNANDES, M. A., AND LOPES, C. R. Automatic student modeling in adaptive educational systems through probabilistic learning style combinations: a qualitative comparison between two innovative stochastic approaches. *Journal of the Brazilian Computer Society 19*, 1 (2013), 43–58.

[31] DREYFUS, H. L. *What computers still can't do: A critique of artificial reason*. MIT press, 1992.

[32] DWORK, C. Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation* (2008), Springer, pp. 1–19.

[33] EPSHTEYN, A., VOGEL, A., AND DEJONG, G. Active reinforcement learning. In *Proceedings of the 25th international conference on Machine learning* (2008), ACM, pp. 296–303.

[34] EVANS, O., STUHLMÜLLER, A., AND GOODMAN, N. D. Learning the preferences of bounded agents. *NIPS Workshop on Bounded Optimality* (2015), 16–22.

[35] EVANS, O., STUHLMÜLLER, A., AND GOODMAN, N. D. Learning the Preferences of Ignorant, Inconsistent Agents. *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence* (2016), 323–329.

[36] EVERITT, T., AND HUTTER, M. Avoiding wireheading with value reinforcement learning. In *Artificial General Intelligence*. Springer, 2016, pp. 12–22.

[37] FINN, C., TAN, X. Y., DUAN, Y., DARRELL, T., LEVINE, S., AND ABBEEL, P. Deep spatial autoencoders for visuomotor learning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)* (2016), IEEE, pp. 512–519.

[38] GONZALES, M. J., CHEUNG, V. C., AND RIEK, L. D. Designing collaborative healthcare technology for the acute care workflow. In *Proceedings of the 9th International Conference on Pervasive Computing Technologies for Healthcare* (2015), ICST (Institute for Computer Sciences, Social-Informatics and , pp. 145–152.

[39] GORDON, G., AND BREAZEAL, C. Bayesian active learning-based robot tutor for children's word-reading skills. In *AAAI* (2015), pp. 1343–1349.

[40] HADFIELD-MENELL, D., RUSSELL, S. J., ABBEEL, P., AND DRAGAN, A. Cooperative inverse reinforcement learning. In *Advances in neural information processing systems* (2016), pp. 3909–3917.

[41] HAYES, C. J., MOOSAEI, M., AND RIEK, L. D. Exploring implicit human responses to robot mistakes in a learning from demonstration task. In *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)* (2016), IEEE, pp. 246–252.

[42] HERMSEN, S., FROST, J., RENES, R. J., AND KERKHOF, P. Using feedback through digital technology to disrupt and change habitual behavior: a critical review of current literature. *Computers in Human Behavior 57* (2016), 61–74.

[43] HOECKELMANN, M., RUDAS, I. J., FIORINI, P., KIRCHNER, F., AND HAIDEGGER, T. Current capabilities and development potential in surgical robotics. *International Journal of Advanced Robotic Systems 12*, 5 (2015), 61.

[44] IQBAL, T., MOOSAEI, M., AND RIEK, L. D. Tempo adaptation and anticipation methods for human-robot teams. *RSS, Planning for HRI: Shared Autonomy and Collab. Robotics Work* (2016).

[45] IQBAL, T., RACK, S., AND RIEK, L. D. Movement coordination in human–robot teams: a dynamical systems approach. *IEEE Transactions on Robotics 32*, 4 (2016), 909–919.

[46] IQBAL, T., AND RIEK, L. D. A method for automatic detection of psychomotor entrainment. *IEEE Transactions on affective computing 7*, 1 (2016), 3–16.

[47] IQBAL, T., AND RIEK, L. D. Human-robot teaming: Approaches from joint action and dynamical systems. *Humanoid Robotics: A Reference* (2019), 2293–2312.

[48] JADERBERG, M., MNIH, V., CZARNECKI, W. M., SCHAUL, T., LEIBO, J. Z., SILVER, D., AND KAVUKCUOGLU, K. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397* (2016).

[49] JEONG, S., LOGAN, D. E., GOODWIN, M. S., GRACA, S., O'CONNELL, B., GOODENOUGH, H., ANDERSON, L., STENQUIST, N., FITZPATRICK, K., ZISOOK, M., ET AL. A social robot to mitigate stress, anxiety, and pain in hospital pediatric care. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts* (2015), ACM, pp. 103–104.

[50] KIDD, C. D., AND BREAZEAL, C. Robots at home: Understanding long-term human-robot interaction. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on* (2008), IEEE, pp. 3230–3235.

[51] KIELHOFNER, G. *Conceptual foundations of occupational therapy practice*. FA Davis, 2009.

[52] KING, R. D., ROWLAND, J., OLIVER, S. G., YOUNG, M., AUBREY, W., BYRNE, E., LIAKATA, M., MARKHAM, M., PIR, P., SOLDATOVA, L. N., ET AL. The automation of science. *Science 324*, 5923 (2009), 85–89.

[53] KING, R. D., WHELAN, K. E., JONES, F. M., REISER, P. G., BRYANT, C. H., MUGGLETON, S. H., KELL, D. B., AND OLIVER, S. G. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature 427*, 6971 (2004), 247.

[54] KUBOTA, A., PETERSON, E. I., RAJENDREN, V., KRESS-GAZIT, H., AND RIEK, L. D. Jessie: Synthesizing social robot behaviors for personalized neurorehabilitation and beyond. *ACM/IEEE International Conference on Human-Robot Interaction* (2020), 1–8.

[55] LANGE, B., CHANG, C.-Y., SUMA, E., NEWMAN, B., RIZZO, A. S., AND BOLAS, M. Development and evaluation of low cost game-based balance rehabilitation tool using the microsoft kinect sensor. In *Engineering in medicine and biology society, EMBC, 2011 annual international conference of the IEEE* (2011), IEEE, pp. 1831–1834.

[56] LEE, H. R., AND RIEK, L. D. Reframing assistive robots to promote successful aging. *ACM Transactions on Human-Robot Interaction (THRI) 7*, 1 (2018), 11.

[57] LEVINE, S., AND KOLTUN, V. Continuous inverse optimal control with locally optimal examples. *arXiv preprint arXiv:1206.4617* (2012).

[58] LI, A. X., FLORENDO, M., MILLER, L. E., ISHIGURO, H., AND SAYGIN, A. P. Robot form and motion influences social attention. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction* (2015), ACM, pp. 43–50.

[59] LIPTON, Z. C., AZIZZADENESHELI, K., KUMAR, A., LI, L., GAO, J., AND DENG, L. Combating reinforcement learning's sisyphean curse with intrinsic fear. *arXiv preprint arXiv:1611.01211* (2016).

[60] LUDDEN, G. D., VAN ROMPAY, T. J., KELDERS, S. M., AND VAN GEMERT-PIJNEN, J. E. How to increase reach and adherence of web-based interventions: a design research viewpoint. *Journal of medical Internet research 17*, 7 (2015).

[61] LUXTON, D. D., AND RIEK, L. D. Artificial intelligence and robotics in rehabilitation.

[62] MAJIDI, C. Soft robotics: a perspectivecurrent trends and prospects for the future. *Soft Robotics 1*, 1 (2014), 5–11.

[63] MILLIGAN, K., AND WISE, D. A. Introduction and summary to" social security and retirement around the world: Historical trends in mortality and health, employment, and disability insurance participation and reforms". In *Social Security Programs and Retirement around the World: Historical Trends in Mortality and Health, Employment, and Disability Insurance Participation and Reforms*. University of Chicago Press, 2012, pp. 1–39.

[64] MITZNER, T. L., CHEN, T. L., KEMP, C. C., AND ROGERS, W. A. Identifying the potential for robotics to assist older adults in different living environments. *International journal of social robotics 6*, 2 (2014), 213–227.

[65] MOHARANA, S., PANDURO, A. E., LEE, H. R., AND RIEK, L. D. Robots for joy, robots for sorrow: Community based robot design for dementia caregivers. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)* (2019), IEEE, pp. 458–467.

[66] MOHR, D. C., SCHUELLER, S. M., MONTAGUE, E., BURNS, M. N., AND RASHIDI, P. The behavioral intervention technology model: an integrated conceptual and technological framework for ehealth and mhealth interventions. *Journal of medical Internet research 16*, 6 (2014).

[67] MOOSAEI, M., DAS, S. K., POPA, D. O., AND RIEK, L. D. Using facially expressive robots to calibrate clinical pain perception. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction* (2017), ACM, pp. 32–41.

[68] NG, A., AND RUSSELL, S. Algorithms for inverse reinforcement learning. *Icml* (2000).

[69] NG, A. Y., HARADA, D., AND RUSSELL, S. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML* (1999), vol. 99, pp. 278–287.

[70] NICHOLSON, N. R. A review of social isolation: an important but underassessed condition in older adults. *The journal of primary prevention 33*, 2-3 (2012), 137–152.

[71] NOTTHOFF, N., AND CARSTENSEN, L. L. Positive messaging promotes walking in older adults. *Psychology and aging 29*, 2 (2014), 329.

[72] ORSEAU, L., AND ARMSTRONG, M. Safely interruptible agents.

[73] PADOY, N., MATEUS, D., WEINLAND, D., BERGER, M.-O., AND NAVAB, N. Workflow monitoring based on 3d motion features. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on* (2009), IEEE, pp. 585–592.

[74] PAPERNOT, N., MCDANIEL, P., JHA, S., FREDRIKSON, M., CELIK, Z. B., AND SWAMI, A. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on* (2016), IEEE, pp. 372–387.

[75] PURNELL, T. S., CALHOUN, E. A., GOLDEN, S. H., HALLADAY, J. R., KROK-SCHOEN, J. L., APPELHANS, B. M., AND COOPER, L. A. Achieving health equity: closing the gaps in health care disparities, interventions, and research. *Health Affairs 35*, 8 (2016), 1410–1415.

[76] QUARTA, D., POGLIANI, M., POLINO, M., MAGGI, F., ZANCHETTIN, A. M., AND ZANERO, S. An experimental security analysis of an industrial robot controller. In *2017 38th IEEE Symposium on Security and Privacy (SP)* (2017), IEEE, pp. 268–286.

[77] RAMACHANDRAN, D., AND AMIR, E. Bayesian inverse reinforcement learning. *Urbana 51*, 61801 (2007), 1–4.

[78] RATLIFF, N. D., BAGNELL, J. A., AND ZINKEVICH, M. A. Maximum margin planning. In *Proceedings of the 23rd international conference on Machine learning* (2006), ACM, pp. 729–736.

[79] RIBES, A., CERQUIDES, J., DEMIRIS, Y., AND LOPEZ DE MANTARAS, R. Active learning of object and body models with time constraints on a humanoid robot. Institute of Electrical and Electronics Engineers.

[80] RIEK, L. D. The social co-robotics problem space: Six key challenges. *Robotics Challenges and Vision (RCV2013)* (2014).

[81] RIEK, L. D. Healthcare robotics. *Communications of the ACM 60*, 11 (2017), 68–78.

[82] ROSS, S., GORDON, G., AND BAGNELL, D. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics* (2011), pp. 627–635.

[83] RUSSELL, S., AND NORVIG, P. *Artificial Intelligence: A modern approach*. Prentice Hall, 1995.

[84] RUST, J. *Do people behave according to Bellman's principle of optimality?* Hoover Institution, Stanford University, 1992.

[85] SAUNDERS, J., SYRDAL, D. S., KOAY, K. L., BURKE, N., AND DAUTENHAHN, K. 'Teach Me-Show Me'-End-User Personalization of a Smart Home and Companion Robot. *IEEE Transactions on Human-Machine Systems 46*, 1 (feb 2016), 27–40.

[86] SETTLES, B. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning 6*, 1 (2012), 1–114.

[87] SHANKAR, A., MCMUNN, A., BANKS, J., AND STEPTOE, A. Loneliness, social isolation, and behavioral and biological health indicators in older adults. *Health Psychology 30*, 4 (2011), 377.

[88] SHI, H.-J., NAKAMURA, K., AND TAKANO, T. Health values and health-information-seeking in relation to positive change of health practice among middle-aged urban men. *Preventive Medicine 39*, 6 (2004), 1164–1171.

[89] SOARES, N., AND FALLENSTEIN, B. Agent foundations for aligning machine intelligence with human interests: a technical research agenda. In *The Technological Singularity*. Springer, 2017, pp. 103–125.

[90] STAUDER, R., OKUR, A., PETER, L., SCHNEIDER, A., KRANZFELDER, M., FEUSSNER, H., AND NAVAB, N. Random forests for phase detection in surgical workflow analysis. In *International Conference on Information Processing in Computer-Assisted Interventions* (2014), Springer, pp. 148–157.

[91] SUTTON, R. S., AND BARTO, A. G. Reinforcement learning: an introduction. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council 9*, 5 (1998).

[92] SYRDAL, D. S., KOAY, K. L., WALTERS, M. L., AND DAUTENHAHN, K. A personalized robot companion?-the role of individual differences on spatial preferences in hri scenarios. In *Robot and Human interactive Communication, 2007. RO-MAN 2007. The 16th IEEE International Symposium on* (2007), IEEE, pp. 1143–1148.

[93] TALL, D. *How humans learn to think mathematically: Exploring the three worlds of mathematics*. Cambridge University Press, 2013.

[94] TAMURA, T., YONEMITSU, S., ITOH, A., OIKAWA, D., KAWAKAMI, A., HIGASHI, Y., FUJIMOOTO, T., AND NAKAJIMA, K. Is an entertainment robot useful in the care of elderly people with severe dementia? *The Journals of Gerontology Series A: Biological Sciences and Medical Sciences 59*, 1 (2004), M83–M85.

[95] TAULBEE, L. R., AND FOLSOM, J. C. Reality orientation for geriatric patients. *Psychiatric Services 17*, 5 (1966), 133–135.

[96] TAYLOR, A., LEE, H., KUBOTA, A., AND RIEK, L. Coordinating clinical teams: Using robots to empower nurses to stop the line. *Proceedings of the ACM Conference on Computer Supported Collaborative Work* (2019).

[97] TAYLOR, J., YUDKOWSKY, E., LAVICTOIRE, P., AND CRITCH, A. Alignment for advanced machine learning systems. *Machine Intelligence Research Institute* (2016).

[98] THE AMERICAN OCCUPATIONAL THERAPY ASSOCIATION, I. What is occupational therapy? https://www.aota.org/Conference-Events/OTMonth/what-is-OT.aspx. Accessed: 2018-12-23.

[99] THOMAZ, A. L., HOFFMAN, G., AND BREAZEAL, C. Experiments in socially guided machine learning: understanding how humans teach. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction* (2006), ACM, pp. 359–360.

[100] TORABI, F., WARNELL, G., AND STONE, P. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954* (2018).

[101] WIRTH, C., AND FÜRNKRANZ, J. Preference-Based Reinforcement Learning : A preliminary survey. *ECML PKDD 2013 - Workshop on Reinforcement Learning from Generalized Feedback: Beyond Numeric Rewards* (2013).

[102] WOODWORTH, B., FERRARI, F., ZOSA, T. E., AND RIEK, L. D. Preference learning in assistive robotics: Observational repeated inverse reinforcement learning. In *Machine Learning for Healthcare Conference* (2018), pp. 420–439.

[103] WOOLF, B. P. *Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning*. Morgan Kaufmann, 2010.

[104] YBARRA, M. L., HOLTROP, J. S., PRESCOTT, T. L., AND STRONG, D. Process evaluation of a mhealth program: Lessons learned from stop my smoking usa, a text messaging-based smoking cessation program for young adults. *Patient education and counseling 97*, 2 (2014), 239–243.

[105] YOSINSKI, J., CLUNE, J., NGUYEN, A., FUCHS, T., AND LIPSON, H. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579* (2015).

[106] ZHANG, S., YAO, L., AND SUN, A. Deep learning based recommender system: A survey and new perspectives. *arXiv preprint arXiv:1707.07435* (2017).

[107] ZHANG, Y. Multi-task active learning with output constraints. In *AAAI* (2010).

[108] ZOSA, T. E., AND YU, M., Jul 2017.