Machine Learning for Interactive Systems and Robots: A Brief Introduction*

Heriberto Cuayáhuitl
Heriot-Watt University
School of Mathematical and Computer Sciences
Edinburgh, United Kingdom
h.cuayahuitl@hw.ac.uk

Martijn van Otterlo
Radboud University Nijmegen
Artificial Intelligence
Nijmegen, The Netherlands
m.vanotterlo@donders.ru.nl

Nina Dethlefs
Heriot-Watt University
School of Mathematical and Computer Sciences
Edinburgh, United Kingdom
n.s.dethlefs@hw.ac.uk

Lutz Frommberger
University of Bremen
Cognitive Systems Group
Bremen, Germany
lutz@informatik.uni-bremen.de

ABSTRACT

Research on interactive systems and robots, i.e. interactive machines that *perceive, act* and *communicate*, has applied a multitude of different machine learning frameworks in recent years, many of which are based on a form of *reinforcement learning* (RL). In this paper, we will provide a brief introduction to the application of machine learning techniques in interactive learning systems. We identify several dimensions along which interactive learning systems can be analyzed. We argue that while many applications of interactive machines seem different at first sight, sufficient commonalities exist in terms of the challenges faced. By identifying these commonalities between (learning) approaches, and by taking interdisciplinary approaches towards the challenges, we anticipate more effective design and development of sophisticated machines that *perceive, act* and *communicate* in complex, dynamic and uncertain environments.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Learning, Language, Robotics

General Terms

Algorithms, Theory

*This research was supported by the EC FP7 programme under grant agreement no. 287615 (PARLANCE), and the German Research Foundation (DFG) through the Transregional Collaborative Research Center SFB/TR 8 "Spatial Cognition".

MLIS'13, August 04 2013, Beijing, China

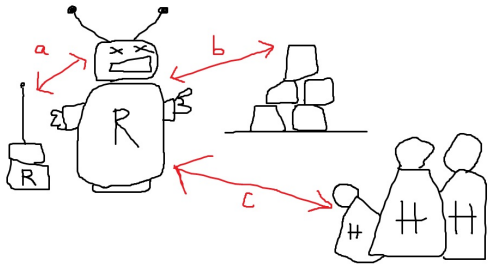

Figure 1: An interactive system, interacting with other robots (a), the world (b), or with humans (c).

1. MOTIVATION

Intelligent systems or robots often learn through continuous interaction with their environment. We define an *interactive machine* here as any entity which learns through interacting with its (real or virtual) physical world, humans and/or other machines. This principle is illustrated in Figure 1, which shows the different ways of interaction that a machine can use for learning. It depicts an interactive system (e.g. a robot) that can interact with one or more other robots, with the world, and with humans. Interaction in such scenarios is seen as a two-way causal relationship in which an agent can *observe* the other end in the interaction, and in which it can *act* as part of the interaction. The resulting observation-action loop can be enriched with *feedback* that can be given to any interaction partner, at any moment. Feedback may include explicit signals being given to the interactive system (experienced through its observations) or may come from *intrinsic* sources, such as motivation, intentions or goals.

Given the many ways interaction can take place, and given the many types of feedback one may apply, *learning* is vital to *optimize interaction patterns*. Even though many robotic systems can be scripted or programmed to behave just as expected, the rich nature of interaction with the physical world, or with humans, demands flexible, adaptive solutions to deal with dynamic, previously unknown, or highly stochastic domains. Learning in interactive systems typically generalizes standard supervised learning settings (e.g. classification). We can adapt a general definition of learning (taken from a classical machine learning textbook [62]) towards: *"A machine can therefore be said to learn from interactions in a particular class of tasks, if its performance improves with the given interactions over time"*.

In interactive systems, learning settings generalize to interaction with other humans or robots, to teaching and learning settings, to complex interaction with the physical world, and to observations being produced by rich sensors such as RGB-3D cameras and stereo microphones. Examples of such rich settings include, but are not limited to:

- a robot that may learn to coordinate its speech with its actions, taking into account visual feedback during their execution;
- an autonomous vehicle (a car, a wheelchair, etc.) that may learn to coordinate its acceleration and steering behaviour depending on observations of obstacles;
- a team of robots playing soccer that may learn to coordinate their ball kicks depending on the dynamic locations of their opponents;
- a mobile robot that may interactively learn from human guidance how to manipulate objects and move through a building, based on human feedback using language, gestures and interactive dialogue.
- a multimodal smart phone can adapt its input and output modalities to the userÕs goals, workload and surroundings.

While intelligent machines can thus interact with their environment in different ways by perceiving, acting and communicating, they often face a challenge in how to bring these different concepts together in a systematic and unified way. Originally, the field of artificial intelligence (AI) was concerned with so-called "complete" systems, in which perception, action, memory, reasoning mechanisms and much more were studied as a whole. However, in recent decades, many specialized fields have appeared in which *performance* on a specialized task was valued, for example natural language processing or object recognition. Such specialized systems are very good at doing one thing, but have lost the connection to complete AI systems displaying truly *general* intelligence.[1] Several pleas, such as the one by Nilsson [67], have been given to go back to the original goals of AI: building general intelligent systems in which many specialized algorithms are integrated. Another, yet related, movement in AI concerned *new*, or *embodied*, AI (see for example the book by Pfeifer and Scheier [73]). This movement is based on the idea that for intelligence to develop, a *physical body* is needed, and it is equally useful for making some tasks simpler because the world-body-mind interaction can be exploited in various ways.

[1]See also the recent conferences on *artificial general intelligence* (AGI) http://agi-conference.org/

In this paper we argue in a similar way for more integration of existing knowledge and expertise, yet now in a more constrained context of machine learning. Namely, one of the main reasons for the existing gap in adaptive interactive systems is the fact that the core concepts in perception, action and communication are typically studied by different communities: computer vision, robotics, natural language processing, human-computer interaction communities, and cognitive science, among others, without much interchange between them. Across communities, the machine learning field has provided us with a rich set of computational methods to improve the individual performance of perception, action and communication components in interactive systems and robots. However, learning systems that encompass multiple of these concepts in a unified and principled way are still rare. As machine learning lies at the core of several communities, we argue that it can act as a unifying factor in bringing the communities closer together. Closer collaboration could be beneficial for practical applications in robotics, human-robot interaction and intelligent interfaces. Equally important, it encourages theoretical advances in adaptive sensorimotor and perception-action loops in general cognition, and understanding how state-of-the-art approaches in each of the disciplines can be combined to form generally interactive intelligent systems.

In the following, we will identify several important aspects of interactive machines and briefly survey some of the most important learning paradigms that have been applied. Note that we do not claim to be exhaustive, and that there are numerous works we cannot mention in this limited space. We also highlight some common opportunities and challenges across paradigms and argue for a more unified perspective on machine learning for interactive systems and robots.

2. DIMENSIONS OF ADAPTIVE, INTERACTIVE MACHINES

Interactive machines come with different capabilities and tasks. It is useful to categorize adaptive interactive systems along a number of dimensions. These include, but are not limited to:

1. *Physical presence:* Does the machine have a (real) physical presence or is it virtual? (an autonomous car versus a driving simulator, e.g.)
2. *Situatedness:* Does the machine act in a (virtual or real) spatial environment which undergoes dynamic changes and can include other entities?
3. *Embodiment:* Does the machine have a (virtual or physical) body which might even resemble human anatomy (e.g. as a humanoid)?
4. *Modalities:* Is the machine uni-modal or does it have multiple input and output channels?
5. *Conversational capability:* Can the machine engage in conversations with humans or other machines? (e.g., a spoken dialogue system or conversational robot)
6. *Affectiveness:* Does the machine display a consistent personality or does it try to adapt its behaviour to the (changing) state of mind of its user? Basically, can it read and/or communicate emotional states?
7. *Flow of interaction:* Is the interaction structured in any way (e.g. using turn-taking), can either partner in the interaction take the initiative, and which way can information and actions flow?

8. *Roles:* Whenever adaptation is part of the interaction, which roles (teacher, apprentice, student, solo-exploring mode, etc.) do partners have and how is feedback used to change the interaction?
9. *Task and team structure:* What is the role of the interactive system in the global system (e.g. equal partners, master-slave setup, advisor, etc.) and what does the system need to accomplish? (i.e. what is the global performance measure?)

The first three dimensions are basically about the role of the system in a "real", physical context. Being *embodied* and *embedded in* an environment can have a huge influence on the types of interactions. The next three dimensions are about communication with the environment, such as the people within it. The last three dimensions – the flow of interaction, aspects of roles, and task and team structure – are about *what* is actually being optimized in the interaction. Learning in interactive systems, the focus of this paper, essentially optimizes the interaction patterns relative to some goal, some task, and some other intelligent components in the system, and a key aspect is *how* feedback is used in the global system to actually do that.

Categorizing interactive machines along these dimensions gives rise to an enormous amount of different interaction patterns, and equally to many different learning settings. For example, a humanoid robot that moves through a dynamic environment and solves tasks by interacting with human instructors exhibits many capabilities that can be categorized along these axes. We leave it to future work to fully characterize these systems in a more rigorous ontology. Note that whereas physical robots are typical examples, equally fitting examples include for example Apple's personal assistant SIRI on a mobile phone.

As said, learning is a vital component of any adaptive interactive system or robot. The following section will give an overview of the most important learning techniques applied to interactive machines.

3. MACHINE LEARNING FRAMEWORKS

Adaptive interactive systems can optimize their behavior in various ways. Possibilities range from a robot doing unsupervised generalization over objects based on camera images, to multiple robots learning how to coordinate a physical task (e.g. carrying a table) from observing humans doing the same thing. Important aspects here are (i) the task to be optimized, (ii) the type of feedback and how it is given to the learning system, and (iii) the modalities used in the system (e.g. dialogues, camera sensors, haptic feedback, etc.).

The typical machine learning task in interactive systems is that of learning a *function* from *inputs* (e.g. state features) to *outputs* (e.g. actions, class labels, preferences, etc.). Feedback on the current behavior of the system can be given in terms of evaluative corrections (e.g. "this behavior is quite good"), correct answers (e.g. "this is the wrong classification; it should have been..."), or generally in terms of numerical rewards which can be optimized (e.g. "every time the robot performs the right answer, it gets +10 reward").

3.1 Type and Amount of Feedback

From a technical point of view, learning frameworks differ in the amount and the way they process *feedback*, and, from a contextual point of view, where the feedback is coming from. Starting from this, we can place those learning paradigms in between the extremes: classical *supervised learning* and *unsupervised learning* (see also [33]). All these variants have their specific purpose and functional role in interactive systems. In the following, we will give a short overview. Note that these descriptions concern typical instances, whereas practically (and conceptually) many combinations can (and some have been) be proposed.

3.1.1 Supervised Learning

Generally speaking, supervised learning can be used whenever it comes to the task of classifying data. Consider a data set of the form: $\mathcal{D} = \{(x_1, y_1), ..., (x_N, y_N)\}$, where x_i are n-dimensional vectors of features and y_j are class labels. A supervised learning algorithm induces a function $h : X \to Y$, where X is the input space (*unlabelled instances*) and Y is the output space (*labels*). The function h is known as a *classifier* when Y is discrete and a *regressor* when Y is continuous. It is among a space of functions $H = \{h|h : X \to Y\}$, and x can be labelled as

$$h(x) = \arg\max_y f(x, y), \tag{1}$$

where f is a scoring function $f : X \times Y \to \mathbb{R}$. This function can be induced through a number of different algorithms such as decision trees, neural networks, Bayes nets, instance-based methods, linear regression, and support vector machines (SVMs), among others [9, 55].

3.1.2 Semi-Supervised Learning

The success of supervised learning depends on a usually very large set of labelled data, which, in many cases, is not available in sufficient quality or amount. Semi-supervised learning algorithms address this problem by using a small amount of labelled data and a large amount of unlabelled data. The aim is (a) to improve the learning accuracy over supervised methods and (b) to reduce the expense in data labelling [116]. Examples of semi-supervised learning methods are generative models [66], transductive SVMs [47], self-training [112], and co-training [10].

3.1.3 Reinforcement Learning

While supervised methods provide a direct feedback to a given input, reinforcement learning (RL) provides a kind of indirect feedback based on rewards for the result of the interaction of a system with its environment, and the aim is to maximize long-term numerical rewards (see [48, 89]). This can be seen as a very weak form of supervised learning, where not the situation itself is rated, but the impact of an action taken towards an overall goal. Thus, reinforcement learning by design complies very well with interactive systems, as knowledge gain is a result of interaction itself.

The RL framework basically uses the formalism of Markov Decision Processes (MDPs). An MDP consists of a finite set of states $S = \{s_i\}$, a finite set of actions $A = \{a_j\}$, a probabilistic state transition function $T = P(s'|s, a)$, and a reward function $R(s'|s, a)$ that rewards the agent for choosing action a in state s (at time t) and transitioning to state s'. Solving an MDP means finding a function $\pi : S \to A$ defined as

$$\pi^*(s) = \arg\max_{a \in A} Q^*(s, a), \tag{2}$$

where the Q-function specifies cumulative rewards for each state-action pair. The policy function π is the basis for

action-selection, and the optimal function Q^* can be induced by reinforcement learning algorithms [89, 93, 110].

While the MDP model offers a formal framework for optimizing the behaviour of interactive systems and robots, its practical application is affected by several limitations. These include large search spaces (for complex domains), partial observability, unknown state transitions and reward functions (all due to uncertainty), slow learning (due to infinite visits to state-action pairs), and several others. However, with recent advances in RL, the use of RL for interactive systems and robots is strongly increasing [110].

There are several other ways RL has been extended with other forms of machine learning to tackle such limitations. The relationship between supervised and reinforcement learning has addressed scalability and model learning of MDPs. On the one hand, since tabular policies (i.e. representing the policy with a lookup table) are hard to scale up, they have been replaced by *function approximators* such as decision and regression trees [75, 28], neural networks [98, 78], linear-based methods [93], and policy gradient methods [90, 72]. In addition, *hierarchical learning* divides a problem into unified sub-problems for accelerating learning and scaling up to more complex problems [7, 20, 51]. On the other hand, supervised learning has been used to estimate the state transition function and reward functions while the agent interacts with its environment. This is known as *model-based learning* [4, 74], in contrast to *model-free learning* which does not require learning a transition and reward function. Model-based learning plays a crucial role in systems and robots that learn from interaction [44].

While little attention has been devoted to semi-supervised RL, one exception combines RL with transductive SVMs in order to induce MDP-based polices with a reward function of the form $R = w^T \phi(s)$ with state features ϕ and weights w estimated from observed behaviour [105].

3.1.4 Unsupervised Learning

In contrast to supervised learning approaches, unsupervised learning algorithms make use of unlabelled training examples $\mathcal{D} = \{x_1, ..., x_n\}$ and consider the labels unknown (i.e. they are *hidden variables*). Thus, the task of an unsupervised learning algorithm is to find hidden structure in unlabelled data sets. Unsupervised learning approaches include clustering, statistical modelling, dimensionality reduction, and unsupervised neural networks. Clustering is the task of partitioning the input data into maximally homogeneous groups (also called *clusters*) [108]. Statistical modelling is used to estimate probability distributions such as $P(x_t|x_1, ..., x_{t-1})$, given a new input x_t and its previous inputs [40]. Dimensionality reduction is the task of finding a lower dimensional representation of the n-dimensional vectors of features in the input space [34]. Unsupervised neural networks are used to learn representations of the input in order to capture salient characteristics at different levels of granularity, e.g. deep learning algorithms [8].

3.2 The Origin of Experience and Feedback

In addition to the *type* of feedback (creating a spectrum ranging from supervised to unsupervised learning), one can distinguish various *sources* of feedback. In addition, more general learning *experiences* (e.g. observations, learning samples, etc.) can come from different origins. Here we briefly review several options.

3.2.1 Transfer Learning / Multi-Task Learning

Transfer and multi-task learning are closely related concepts that aim at *learning to learn* [100] in a way that they try to improve the learning performance in a task based on previous learning efforts. Usually, we have two different data sets for source and target data, $\mathcal{D}^s = \{(x_1^s, y_1^s), ..., (x_n^s, y_n^s)\}$ and $\mathcal{D}^t = \{(x_1^t, y_1^t), ..., (x_m^t, y_m^t)\}$; $\mathcal{D}^s \neq \mathcal{D}^t$. Source data may come from multiple data sets. The difference in the data sets may be in the feature vectors x_i, in the labels y_j, or in both. Thus, given the source data $\mathcal{D}^s$ and its corresponding scoring function f^s (see Equation 1), a transfer learner aims to help learning the target scoring function $f^t : X^t \times Y^t \rightarrow \mathbb{R}$. This reduces the amount of labelled data in a target task and avoids learning from scratch (for faster learning). Supervised and unsupervised learning methods, as described above, have been used to transfer knowledge of feature vectors and parameters of the scoring function from the source task(s) to the target related task [71]. A generalization of semi-supervised learning (referred to as *self-taught learning*) learns high-level representations of the input space from labelled and unlabelled data from different but similar domains, and uses them for classifications tasks [76].

In reinforcement learning, transfer learning has been established through a multitude of approaches to transfer knowledge from a source MDP M^s to a target MDP M^t. The type of knowledge to transfer can be derived from the states S, actions A, state transition function T, reward function R, and/or (partial) policies optimal π^* [96]. We can roughly distinguish between *intra-domain* transfer (where only R differs in M^s and M^t) and *cross-domain* transfer (where also S, A, and T can differ). For intra-domain transfer, all kinds of temporal abstraction methods (e.g. options [92] or MAXQ [26]) can be used for knowledge transfer. The same holds for *policy reuse* [30, 22, 81], which makes use of previously learned policies in the same domain. Cross-domain transfer methods include the use of abstract rules to transfer value functions [103, 97], create shaping rewards [54], hierarchical reinforcement learning [101, 85], or exploiting state space abstraction by creating abstract skills [53] or initialization of Q-functions [36]. There are many more approaches to transfer in RL contexts [57]. What all of these approaches have in common is that they rely on some similarity of source and target task. This is mostly given by a task mapping function introduced outside of the learning task or exploitation of external concepts (such as relations or agent spaces) that are defined a-priori. The big challenge for interactive systems is to notice and exploit this similarity on their own [60], e.g., by identifying relevant state variables [86]. For such tasks, the representation of the state space is of critical relevance, and the importance of representation for knowledge reuse is widely acknowledged [106, 35].

3.2.2 Active Learning

In *active learning* settings, an interactive system has influence on the learning experiences it gets (using prior experience, exploration, knowledge or other means). An active learning algorithm makes use of three data sets: labelled examples $\mathcal{D}^l$, unlabelled examples $\mathcal{D}^u$, and chosen examples $\mathcal{D}^c$. The last data set is built in an interactive fashion by the learning algorithm who queries a human annotator for labels of those examples it is most uncertain of. A number of methods have been investigated for choosing the examples to label [84]; e.g. in uncertainty sampling an entropy-based

sampling strategy is defined as

$$x^* = \arg\max_x - \sum_i P(y_i|x) \log P(y_i|x), \quad (3)$$

where x^* is the best query and y_i are all the possible labels.

Active learning has been combined with RL for determining the sensitivity of the optimal policy to changes in state transitions and rewards. Active RL is used to explore regions of the state-action space where the optimal policy has the most uncertainty [27]. In addition, active learning has been investigated for reward function estimation, where the RL agent queries a demonstrator for examples at specific states [61]. These investigations have led to more efficient learning than using passive reinforcement learning.

3.2.3 Social Learning Strategies

Interactive learning means learning in interaction *with*, or *in the context of* other intelligent beings such as humans and robots. The interactive system can function as a *teacher*, a *student*, *an apprentice*, etc. Many forms of *social learning* can be used in the context of interactive systems, such as *imitation*, *copying*, *learning together*, *learning from others* and many other forms found in the social learning literature [77]. Exploiting social partners [15] can speed up (or even make possible) learning a lot and enables the *reuse* of knowledge gained by other intelligent beings (which in that case is similar to transfer learning settings).

One simple extension of RL approaches to interactive systems is to employ human judgement of the quality of behavior as a reward function in a standard RL setting. This was proposed in the *interactive RL* approach [99]. Further application of this idea has led to new insights into RL, for example on the complexity of learning with a social partner [88]. In addition, several approaches are now being investigated to add human-generated reward models into RL, see for example [49]. Alternatively one can also allow the interactive system to ask *questions* in an interactive RL setting to gain more information about task performance [16]. Such an approach is essentially an active learning setting.

Learning from demonstration (LfD) has been framed as a generalization of supervised learning [2]. While a supervised learner is given a set of labelled examples, an LfD algorithm is given example executions $\mathcal{D} = \{s_i, a_i\}$ of a task by a demonstration teacher. Solving an LfD problem means finding a policy for selecting action $a \in A$ in state $s \in S$. A demonstration consists of a sequence of state-action pairs, and a policy can be induced using batch learning or interactive (online) learning. In the former, a set of demonstrations is given to an LfD algorithm offline. In the latter, demonstrations are given incrementally as they become available. Since the quality of the learned policies depends on the quality of the demonstrations, LfD algorithms need to generalize from the provided demonstrations. Some approaches first allow the robot to explore in order to estimate a personal, local environment model (in the form of *affordances*), and subsequently use that model to transform human-generated, video-based demonstration to the robot's own action repertoire [63]. Recently, several algorithms were compared empirically [102] and many RL-based techniques currently exist, for example for robotics [50]. A related idea in RL methods is the *inverse RL* strategy [115], in which the interactive system observes a demonstration and tries to estimate the reward function from that. This way, the agent learns what actually drives the demonstrated behavior (in other words: how can one evaluate the demonstrated behavior as good?).

Model-based reinforcement learning approaches have been used to estimate state transition functions and reward functions for policy learning from demonstrations [5, 43]. Active learning has also played a role in combination with batch learning, in that the demonstrator can be queried about what to do in states with high uncertainty [61].

3.3 Other Relevant Settings

Since this paper is about interactive learning, it is natural to consider learning settings involving multiple agents. So-called *multi-agent learning systems* are used when different entities with different (possibly conflicting) goals need to optimize their interactions [87]. Multi-Agent Reinforcement Learning (MARL) is a generalization of the RL framework to solve multiple MDPs concurrently [14, 69, 68]. An MDP in a multi-agent setting is defined as $M =< S, \bar{A}, T, \bar{R} >$, where $\bar{A} = \{A_1, ..., A_n\}$ is the set of joint actions (one set for each agent i) and $\bar{R} = \{R_1, ..., R_n\}$ is the set of reward functions for each agent. There are three main types of multi-agents: cooperative, independent and adversarial. In contrast to *cooperative agents* that induce a joint learned policy

$$\pi^*(s_t) = \arg\max_{\bar{a}_t \in \bar{A}} Q^*(s_t, \bar{a}_t), \quad (4)$$

independent agents induce policies without joint decisions, i.e. they learn independent Q-functions $Q^*(s, a_i)$, one for each agent i. They are often used as a benchmark for other forms of multi-agent learning. All agents use the same environment states, and the execution of actions affects their shared environment. While independent agents may use separate reward functions, cooperative agents use the same reward function in order to optimize a joint policy $Q : S \times \bar{A} \rightarrow \mathbb{R}$. Although cooperative agents have shown to outperform independent ones [95], they also require the need for coordination of selected actions among agents [12, 56]. This coordination is necessitated by the fact that the effects of each agent-action on the environment depend on the actions taken by the other agents. In addition, coordinated MARL has been investigated in the context of hierarchical RL agents, which often learn to coordinate faster [41, 24]. Finally, *adversarial agents* induce a policy of the form

$$\pi^*(s_t) = \arg\max_{a_t \in \bar{A}} \min_{o_t \in O} Q^*(s_t, a_t, o_t), \quad (5)$$

where o is the action (or joint actions) of the opponent agent(s). The goal is therefore to maximize the rewards of one's own actions while minimizing the rewards of the opponent's actions. Stochastic Markov games have been investigated for adversarial RL [59, 45], which have been useful to learn strategies from opponent agents. They have addressed the notions of continuous states [104] and variable learning rate to improve convergence to optimal policies [13].

Other relevant learning settings include (but are certainly not limited to) for example *preference learning* [37], possibly combined with a RL setting [38]. Active preference-based RL has been suggested which incorporates preferences in learning from demonstration. This work seems promising for optimizing preference-based behaviours in an interactive fashion [1].

Also relevant for interactive systems, especially if they need to communicate high-level knowledge with human com-

panions, is the use of *commonsense, high-level knowledge representation languages* for both learning and reasoning. Many examples exist, also in the RL setting [107].

Another interesting setting may involve learning on a *large set of tasks*, which has so far received little attention. The multi-task learning systems described earlier have addressed only few tasks. Machine learning systems capable of dealing with a large set of tasks remain to be investigated, which may involve the integration of several (if not all) machine learning frameworks described above.

4. ADAPTIVE INTERACTIVE SYSTEMS: EXAMPLES AND CHALLENGES

In an earlier section we have briefly discussed several important dimensions of adaptive interactive systems, and in the previous section we have surveyed some of the main learning paradigms which could be included in such systems. Let us now complete our discussion with some examples that illustrate such notions, and we conclude with an outlook to the MLIS workshop of this year.

4.1 From Virtual Assistants to Interactive Conversational Robots

Imagine a virtual assistant operated by voice that enables dialling, dictation, searching information on the web, managing contacts and agendas, finding locations, and visualizing maps, among others. Nowadays, these kinds of systems are ubiquitous in the form of Apple's *Siri*, Nuance's *Nina* or *Vlingo*. In order to fulfill a wide range of tasks, these assistants make use of a core of language technology components. Briefly, they use speech recognition to extract the user's words, language understanding to extract the hidden meaning of words, interaction management to decide what to do next (in the case of systems beyond a single-query), language generation to generate textual outputs, speech synthesis to generate spoken outputs, and graphical interfaces to visualize content to the user (in the case of systems with multimodal output). Although these components have made use of several forms of machine learning mentioned in the previous section, some forms of learning remain to be investigated further. In particular, learning from demonstration, transfer and multi-task learning, multi-agent learning, preference learning, and large-scale learning have not yet applied to virtual agents.

In addition, though, it is conceivable to develop assistants with further multimodal capabilities, such as machines that require to see and move. An example is the Simon robot [18]. Its task is to learn different configurations of tangram pairs and associate them with meanings. A small round object on top of a larger round object, for example, is a *snowman*. The robot learns such mappings in a supervised fashion from human teachers that provide a range of positive and negative labelled examples. From these, the robot will try to make as many generalizations as possible. For example, presented with a set of objects that differ along all dimensions except for colour, Simon can learn the concept of the colour *red*. Since often the space of possible configurations is very large, though, it has the additional capability of signalling to the human teacher which objects it is still confused about. This is done through active learning: the robot raises learning queries to the human teacher by identifying objects using non-verbal gestures and requests labels for them. This skill can substantially accelerate exploration of the hypothesis space and thereby new learning. While Simon learns online, from real human interactions, many of its behaviours can seem simple at first glance and lack the apparent sophistication displayed by the commercially available virtual assistants discussed above.

In contrast to agents like Simon that learn from relatively few non-verbal examples due to its small world, interactive systems and robots in dialogue domains [58, 114] typically require large training data sets–often too much to provide learning during the course of the interaction. Consequently, most work to-date has relied on training from simulated dialogues [83] with little advances in *online learning* [23]. While some authors have made progress towards learning dialogue behaviour from human-machine interaction, they still rely on some form of simulation or delayed re-training. [11], for example, describe a spoken dialogue system that learns to optimize its non-understanding recovery strategies on-line through interactions with human users based on pre-trained logistic regression models. The system is re-trained every day. [21] present a dialogue system in the navigation domain that is based on hierarchical RL and Bayesian Networks and re-learns its behaviour after each user turn, using indirect feedback from the user's performance. [39] present a spoken dialogue system that uses Gaussian Process-based RL. It learns from binary feedback that users assign explicitly as rewards at the end of each dialogue and that indicate whether users were happy or unhappy with the system's performance. The system is then re-trained after every dialogue. [25] present a robot companion that learns to ask and answer questions, which uses hierarchical RL with dynamic tree-based state representations that can grow during the course of the dialogue. This enables users to take more flexible control of the interaction than in typical hierarchical RL settings. Some of the reasons that online learning has been little explored so far include the large number of training examples required, and the subjective nature of human assessment of dialogue behaviour. Nonetheless, the progress above leads to interactive systems and robots that *avoid learning from scratch* every time a new system is constructed.

4.2 From Simple Behaviors to Generalized Skills

When it comes to non-verbal behaviors, possibly combined with verbal ones, the number of possibilities for learning behaviors, policies or skills is basically infinite.

Learning low-level behaviors and predictions is receiving attention lately in the area of *sensorimotor learning*, in which basic perception-action interaction loops are learned from large, noisy data streams. An interesting algorithm from the field of RL is *Horde* [91] in which many simple *daemons* learn to predict single pieces (bits, features) of information from a stream of data. Each daemon learns *off-policy*, generalizing value functions using linear value function approximation. An example prediction would be *"what would my speed be after constantly hitting the acceleration button?"* Such adaptive sensorimotor control loops form the basis of any interactive system functioning in the real world.

On a higher level, many forms of interaction could be the subject of an optimization process using learning. Non-embedded, non-embodied types of systems, such as interactive game playing programs, may exhibit many forms of learning. For example, learning how to play games, how

to do that in interaction, and also how learn how to adapt to the human player such that game playing experience is optimized (not too easy, not too difficult, fun, entertaining, challenging). Many examples, ranging from Checkers and Chess, to Go, and to real-time strategy games (RTS) and first-person action shooters (FPS), exist in the literature (see [94] for a survey). Online versions of those systems, in terms of *massively online* internet games (e.g. *World of WarCraft*) may require more sophisticated interactive machines that can handle the real-time, massively multi-player, and social nature of those games. Many forms of social learning would be required here, as well as ways to do *opponent model learning*.

Now, moving towards physical, embedded, embodied systems, we again open up more opportunities and challenges for adaptive, interactive systems. In addition to standard learning settings, *physical* aspects of environments generate new learning settings. For example, the aspect of *proxemics* [64], e.g. the distance between interaction partners, in the interaction becomes very important suddenly. Combining such issues with *gaze* [65] and *looking direction* give rise to whole new problems. For example, people will keep a greater distance between them and a robot if the robot is directly looking at them while the person is moving. On the other hand, gazing behavior can greatly help when looking at semantically meaningful objects in the environment, especially when coupled with a verbal dialogue mentioning these objects. Learning such aspects, and how intentions and plans can be detected from physical movement or gaze (or vice versa) is a very interesting direction for research, both for interpreting behavior as well as perception in general.

In addition to movements and visual input, physical environments also give opportunities to *feel*, i.e. using *tactile* interactions [3]. Learning what a pat on the back means in a given context would greatly benefit overall understanding of social mechanisms. Somewhere in between are visual input patterns of movements that are used to convey meaning: *gestures* [82]. Typically gestures have been studied in the *human-computer interaction* field, but having an intelligent, pysically present interaction partner means that gesture recognition should be paired with gesture *generation* and it should be learned (and taught) both ways. In addition, many communicative gestures are not so much intentionally, but they do follow some (social) conventions [42]. Many of these (social) signals and cues play a vital role in interaction and should become part of any learning setting. These are all interesting directions for research.

Scaling up even more towards general *skills*, a recent trend in general AI, and also interactive systems such as robots, is to scale up using high-level representations (and accompanying learning algorithms). An interesting effort is the *RoboEarth* language [109], which was developed to *transfer* learned knowledge from one robot to another, accross the earth, accross different hardware, and accross different tasks. For example, the high-level skill *fetch-beer* could be subdivided into tasks such as *locate-kitchen*, *find-fridge*, *open-door*, *grab-bottle*, *navigate-to-user*, and so on. Depending on the level of generalization, a robot having mastered this skill could transfer a generalized plan to other robots facing similar requests. Many recent adaptive robot systems are being endowed with high-level *programs* which incorporate enough knowledge about the task, but which also leave open the opportunity to learn or finetune behaviors in the context of changing situations, uncertainty and noise. Many of these systems generalize from simple behaviors towards general skills using more powerful representations. For example [63] generalize *affordance models* first developed for simple situations of a single object (with physical properties such as size) to general situations involving *structured* situations involving many *objects* and *relations* between them. This research combines learning settings for object perception, robot control, grabbing positions, learning from demonstration, dynamics models, and several others; all both for low-level data, and for high-level structures. Such systems are typical examples of adaptive, interactive systems with general intelligence that need to be developed further.

4.3 Some Challenges

To build intelligent interactive systems involving verbal and non-verbal skills, a combination of robust interaction and autonomous learning is required. For this, the following challenges need to be overcome in future research.

1. Interactive robotic systems are currently developed and trained for a specific task and often do not generalize to other tasks. As a consequence, a significant amount of system development is required for the interactive system or robot to take on even closely related tasks.
2. Interactive systems, such as robots, are hindered by several big challenges concerning their learning setting [50]. These include *the curse of X* with X being *dimensionality*, *real-world samples*, *real-world interactions*, *model errors* and *lack of goal specifications*. Progress on all these challenges is needed to acquire truly intelligent, adaptive systems.
3. Typically, these systems are trained in simulation environments and therefore can fail to learn behaviours that are generalizable to the real world.
4. They assume a known and fixed environment, while the real world is partially known and dynamic.
5. They regard a fixed subset of possible sensory inputs, usually tailored to the limited number of tasks to be mastered. When aiming at being adaptive also in the type and aim of interaction, the question of relevant input features, their detection, selection, and representation becomes evident.
6. Nowadays, since interactive systems are (sometimes by far) not capable of exhibiting all desired features for interaction (either in terms of necessary intelligence or in terms of real-time requirements), so-called *Wizard-of-Oz* experiments are conducted [80]. In such experiments parts of the interactive system are being controlled by a human (without possible human interaction partners actually knowing). It is unclear what the consequences of this are for developing truly generaly interactive skills.
7. In addition, *evaluation standards* [113] for interative systems are fairly undeveloped. Using existing standards for *human-computer interaction* may be possible for limited application scenarios, but we also need evaluation metrics for things as *"how well does my robot react to my sloppy teaching?"*, *"how much progress did the system make today in involving people in joyful interactions?"* and *"what is the influence of the physical context on the learning progress of the system so far?"*.

8. They assume a small and fixed knowledge base, which drastically restricts human-machine interactions to small rather than multiple tasks with flexible interchange between them. A potential for future interactive systems are automatically extracted large-scale knowledge bases such as NELL [17].
9. Interactive conversational systems face the long-standing challenge of speech recognition and understanding, especially for distance-based human-robot interaction without the need of headset microphones.
10. Finally, a more wholistic perspective is needed to achieve a principled and seamless integration from different fields such as language, robotics and vision.

4.4 Outlook on MLIS

The Machine Learning for Interactive Systems (MLIS) Workshop Series[2] includes contributions from all areas of perception, action and communication, providing an interdisciplinary perspective with a common thread. Particular topics of interest in 2013 that span through the contributions are more autonomous learning, away from human intervention, and the exploration of new feedback signals.

In terms of perception, [29] present work on spatio-temporal recognition of facial expressions, human activity and hand gestures. The approach is based on finite Beta-Liouville mixture models, which can learn from a small number of parameters and find the best number of mixture components automatically–without requiring hand-specified constraints. Drawing on geometric features, but with a focus on learning object-action relations for object manipulation, [111] infer actions that can be performed on unseen objects using homogeneity analysis. An interactive system, such as a robot performing practical tasks, can in this way transfer its existing knowledge of what actions can be performed on particular objects to new instances, putting it in a position to interact with new environments.

Learning through action in an unknown environment is also addressed by [46]. The authors optimize an agent's behavior directly based on feedback in the form of a user's brain signals, particularly those signals triggered after committing or observing an error. Since such signals provide limited information as to the nature of the error, inverse reinforcement learning is used to infer the user's overall goal. Learning through feedback is also relevant to dialogue. [31] optimize the dialogue behavior of a system based on positive and negative social cues, which serve as additional feedback besides objective rewards. Such signals are available throughout the interaction and allow the agent to learn from multiple sources, objective and social. Finally, evaluation is an important aspect of every interactive system. [32] propose to optimize the interaction time of systems based on a combination of the Keystroke-Level Model and a Markov Decision Process (MDP). An MDP, trained from real user-system interactions, offers the possibility to simulate different conditions, so that first usability tests can be conducted under reduced resources and costs.

The theme of autonomous learning and skill acquisition is also reflected in our invited talks. [79] will discuss interactive systems that learn from raw, real-world input data based on generalization and learning from experiencing success or failure. [52] will give an overview of methods for autonomous skill acquisition that allow robots to acquire general learning skills to transfer knowledge from one task to another. [70] will provide an overview of transfer learning methods and their application to three different tasks: language processing, image classification and Wi-Fi-based localization. [6] will provide a commercial perspective on the development and application of interactive robots and introduce the research carried out at the Aldebaran A-Lab. Finally, [19] will discuss some requirements and progress in applying open knowledge to service robots within the scope of a large set of tasks.

We look forward to inspiring presentations and discussions at MLIS-2013 and beyond!

[2] `http://mlis-workshop.org`

5. REFERENCES

[1] R. Akrour, M. Schoenauer, and M. Sebag. APRIL: Active Preference Learning-Based Reinforcement Learning. In *ECML/PKDD (2)*, pages 116–131, 2012.

[2] B. Argall, S. Chernova, M. M. Veloso, and B. Browning. A Survey of Robot Learning from Demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009.

[3] B. D. Argall and A. G. Billard. A survey of tactile human–robot interactions. *Robotics and Autonomous Systems*, 58(10):1159–1176, 2010.

[4] C. G. Atkeson and J. C. Santamaría. A Comparison of Direct and Model-based Reinforcement Learning. In *ICRA*, pages 3557–3564, 1997.

[5] C. G. Atkeson and S. Schaal. Robot Learning From Demonstration. In *ICML*, pages 12–20, 1997.

[6] J.-C. Baillie. Developmental Robotics at Aldebaran A-Lab. In *Proceedings of the Second Workshop on Machine Learning for Interactive Systems (MLIS)*. ACM ICPS, 2013.

[7] A. Barto and S. Mahadevan. Recent Advances in Hierarchical Reinforcement Learning. *Discrete Event Dynamic Systems: Theory and Applications*, 13(1-2):41–77, 2003.

[8] Y. Bengio. Learning Deep Architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127, 2009.

[9] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[10] A. Blum and T. M. Mitchell. Combining Labeled and Unlabeled Data with Co-Training. In *COLT*, pages 92–100, 1998.

[11] D. Bohus, B. Langner, A. Raux, A. W. Black, M. Eskenazi, and A. I. Rudnicky. Online Supervised Learning of Non-Understanding Recovery Policies. In *SLT*, pages 170–173, 2006.

[12] C. Boutilier. Sequential Optimality and Coordination in Multiagent Systems. In *IJCAI*, pages 478–485, 1999.

[13] M. H. Bowling and M. M. Veloso. Multiagent Learning Using a Variable Learning Rate. *Artif. Intell.*, 136(2):215–250, 2002.

[14] L. Busoniu, R. Babuska, and B. D. Schutter. A Comprehensive Survey of Multiagent Reinforcement Learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 38(2):156–172, 2008.

[15] M. Cakmak, N. DePalma, R. Arriaga, and A. Thomaz. Exploiting social partners in robot learning. *Autonomous Robots*, 29(3-4):309–329, 2010.

[16] M. Cakmak and A. L. Thomaz. Designing robot learners that ask good questions. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, HRI '12, pages 17–24, New York, NY, USA, 2012. ACM.

[17] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. H. Jr., and T. M. Mitchell. Toward an Architecture for Never-Ending Language Learning. In *AAAI*, 2010.

[18] C. Chao, M. Cakmak, and A. Thomaz. Transparent Active Learning for Robots. In *Proceedings of the 5th ACM/IEEE international Conference on Human-Robot Interaction*, HRI '10, 2010.

[19] X. Chen. Open Knowledge for Human-Robot Interaction. In *Proceedings of the Second Workshop on Machine Learning for Interactive Systems (MLIS)*. ACM ICPS, 2013.

[20] H. Cuayáhuitl. *Hierarchical Reinforcement Learning for Spoken Dialogue Systems*. PhD thesis, School of Informatics, University of Edinburgh, January 2009.

[21] H. Cuayáhuitl and N. Dethlefs. Optimizing situated dialogue management in unknown environments. In *INTERSPEECH*, pages 1009–1012, 2011.

[22] H. Cuayáhuitl and N. Dethlefs. Spatially-Aware Dialogue Control Using Hierarchical Reinforcement Learning. *ACM Transactions on Speech and Language Processing*, 7(3):5:1–5:26, 2011.

[23] H. Cuayáhuitl and N. Dethlefs. Dialogue Systems Using Online Learning: Beyond Empirical Methods. In *NAACL-HLT Workshop on Future Directions and Needs in the Spoken Dialog Community: Tools and Data*, SDCTD '12, pages 7–8, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[24] H. Cuayáhuitl and N. Dethlefs. Hierarchical multiagent reinforcement learning for coordinating verbal and non-verbal actions in robots. In *ECAI Workshop on Machine Learning for Interactive Systems (MLIS)*, pages 27–29, Montpellier, France, 2012.

[25] H. Cuayáhuitl, I. Kruijff-Korbayová, and N. Dethlefs. Hierarchical Dialogue Policy Learning using Flexible State Transitions and Linear Function Approximation. In *COLING (Demos)*, pages 95–102, 2012.

[26] T. G. Dietterich. Hierarchical reinforcement learning with the MAXQ value function decomposition. *International Journal of Artificial Intelligence Research*, 13:227–303, 2000.

[27] A. Epshteyn, A. Vogel, and G. DeJong. Active Reinforcement Learning. In *ICML*, pages 296–303, 2008.

[28] D. Ernst, P. Geurts, and L. Wehenkel. Tree-Based Batch Mode Reinforcement Learning. *JMLR*, 6:503–556, 2005.

[29] W. Fan and N. Bouguila. Expectation Propagation Learning of Finite Beta-Liouville Mixtures for Spatio-temporal Object Recognition. In *Proceedings of the Second Workshop on Machine Learning for Interactive Systems (MLIS)*. ACM ICPS, 2013.

[30] F. Fernández and M. M. Veloso. Probabilistic Policy Reuse in a Reinforcement Learning Agent. In *AAMAS*, pages 720–727, 2006.

[31] E. Ferreira and F. Lefèvre. Social Signal and User Adaptation in Reinforcement Learning-based Dialogue Management. In *Proceedings of the Second Workshop on Machine Learning for Interactive Systems (MLIS)*. ACM ICPS, 2013.

[32] L. A. Ferreira, A. A. Masiero, P. T. A. Junior, and R. A. C. Bianchi. Automatic Interface Optimization through Random Exploration of Available Elements. In *Proceedings of the Second Workshop on Machine Learning for Interactive Systems (MLIS)*. ACM ICPS, 2013.

[33] P. Flach. *Machine Learning: The Art and Science of Algorithms that Make Sense of Data*. Cambridge University Press, 2012.

[34] I. K. Fodor. A Survey of Dimension Reduction Techniques. Technical Report UCRL-ID-148494, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, June 2002.

[35] L. Frommberger. *Qualitative Spatial Abstraction in Reinforcement Learning*. Cognitive Technologies. Springer, Berlin Heidelberg, Nov. 2010.

[36] L. Frommberger and D. Wolter. Structural knowledge transfer by spatial abstraction for reinforcement learning agents. *Adaptive Behavior*, 18(6):507–525, 2010.

[37] J. Fürnkranz and E. Hüllermeier. Preference Learning. In *Encycl. of Machine Learning*, pages 789–795. 2010.

[38] J. Fürnkranz, E. Hüllermeier, W. Cheng, and S.-H. Park. Preference-based Reinforcement Learning: A Formal Framework and a Policy Iteration Algorithm. *Machine Learning*, 89(1-2), 2012.

[39] M. Gasic, F. Jurcícek, B. Thomson, K. Yu, and S. Young. On-Line Policy Optimisation of Spoken Dialogue Systems via Live Interaction with Human Subjects. In *ASRU*, pages 312–317, 2011.

[40] Z. Ghahramani. Unsupervised Learning. In *Advanced Lectures on Machine Learning*, pages 72–112, 2003.

[41] M. Ghavamzadeh, S. Mahadevan, and R. Makar. Hierarchical Multi-Agent Reinforcement Learning. *AAMAS*, 13(2):197–229, 2006.

[42] F. Hegel, S. Gieselmann, A. Peters, P. Holthaus, and B. Wrede. Towards a typology of meaningful signals and cues in social robotics. In *RO-MAN, 2011 IEEE*, pages 72–78, 2011.

[43] T. Hester, M. Quinlan, and P. Stone. RTMBA: A Real-Time Model-Based Reinforcement Learning Architecture for Robot Control. In *ICRA*, pages 85–90, 2012.

[44] T. Hester and P. Stone. Learning and using models. In M. Wiering and M. van Otterlo, editors, *Reinforcement Learning: State-of-the-Art*, chapter 4. Springer, 2012.

[45] J. Hu and M. P. Wellman. Multiagent Reinforcement Learning: Theoretical Framework and an Algorithm. In *ICML*, pages 242–250, 1998.

[46] I. Iturrate, J. Omedes, and L. Montesano. Shared Control of a Robot Using EEG-based Feedback Signals. In *Proceedings of the Second Workshop on Machine Learning for Interactive Systems (MLIS)*. ACM ICPS, 2013.

[47] T. Joachims. Transductive Inference for Text Classification using Support Vector Machines. In *ICML*, pages 200–209, 1999.

[48] L. Kaelbling, M. Littman, and A. Moore. Reinforcement Learning: A Survey. *JAIR*, 4:237–285, 1996.

[49] W. B. Knox, B. D. Glass, B. C. Love, W. T. Maddox, and P. Stone. How humans teach agents - a new experimental perspective. *I. J. Social Robotics*, 4(4):409–421, 2012.

[50] J. Kober and J. Peters. Reinforcement learning in robotics: A survey. In M. Wiering and M. van Otterlo, editors, *Reinforcement Learning: State-of-the-Art*, chapter 18. Springer, 2012.

[51] G. Konidaris. *Autonomous Robot Skill Acquisition*. PhD thesis, Department of Computer Science, University of Massachusetts Amherst, May 2011.

[52] G. Konidaris. Robots, Skills, and Symbols. In *Proceedings of the Second Workshop on Machine Learning for Interactive Systems (MLIS)*. ACM ICPS, 2013.

[53] G. Konidaris and A. G. Barto. Efficient Skill Learning using Abstraction Selection. In *IJCAI*, pages 1107–1112, 2009.

[54] G. D. Konidaris and A. G. Barto. Autonomous shaping: Knowledge transfer in reinforcement learning. In *Proceedings of the Twenty Third International Conference on Machine Learning (ICML 2006)*, pages 489–49, Pittsburgh, PA, June 2006.

[55] S. B. Kotsiantis. Supervised Machine Learning: A Review of Classification Techniques. *Informatica (Slovenia)*, 31(3):249–268, 2007.

[56] M. Lauer and M. A. Riedmiller. An Algorithm for Distributed Reinforcement Learning in Cooperative Multi-Agent Systems. In *ICML*, pages 535–542, 2000.

[57] A. Lazaric. Transfer in reinforcement learning: A framework and a survey. In M. Wiering and M. van Otterlo, editors, *Reinforcement Learning: State-of-the-Art*, chapter 5. Springer, 2012.

[58] O. Lemon and O. Pietquin. Machine Learning for Spoken Dialogue Systems. In *INTERSPEECH*, pages 2685–2688, 2007.

[59] M. L. Littman. Markov Games as a Framework for Multi-Agent Reinforcement Learning. In *ICML*, pages 157–163, 1994.

[60] Y. Liu and P. Stone. Value-function-based transfer for reinforcement learning using structure mapping. In *Proceedings Of The National Conference On Artificial Intelligence (AAAI)*, Boston, MA, July 2006.

[61] M. Lopes, F. S. Melo, and L. Montesano. Active Learning for Reward Estimation in Inverse Reinforcement Learning. In *ECML/PKDD (2)*, pages 31–46, 2009.

[62] T. M. Mitchell. *Machine learning*. McGraw Hill series in Computer Science. McGraw-Hill, 1997.

[63] B. Moldovan, P. Moreno, M. van Otterlo, J. Santos-Victor, and L. De Raedt. Learning relational affordance models for robots in multi-object manipulation tasks. In *ICRA*, pages 4373 –4378, May 2012.

[64] J. Mumm and B. Mutlu. Human-robot proxemics: physical and psychological distancing in human-robot interaction. In *Proceedings of the 6th international conference on Human-robot interaction*, HRI '11, pages 331–338, New York, NY, USA, 2011. ACM.

[65] B. Mutlu, T. Kanda, J. Forlizzi, J. Hodgins, and H. Ishiguro. Conversational gaze mechanisms for humanlike robots. *ACM Trans. Interact. Intell. Syst.*, 1(2):12:1–12:33, Jan. 2012.

[66] K. Nigam, A. McCallum, S. Thrun, and T. M. Mitchell. Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning*, 39(2/3), 2000.

[67] N. J. Nilsson. Human-level artificial intelligence? be serious! *AI Magazine*, 2005.

[68] A. Nowe, P. Vrancx, and Y.-M. D. Hauwere. Game theory and

multi-agent reinforcement learning. In M. Wiering and M. van Otterlo, editors, *Reinforcement Learning: State-of-the-Art*, chapter 14. Springer, 2012.

[69] F. A. Oliehoek. Decentralized POMDPs. In M. Wiering and M. van Otterlo, editors, *Reinforcement Learning: State-of-the-Art*, chapter 15. Springer, 2012.

[70] S. J. Pan. Transfer Learning with Applications on Text, Sensors and Images. In *Proceedings of the Second Workshop on Machine Learning for Interactive Systems (MLIS)*. ACM ICPS, 2013.

[71] S. J. Pan and Q. Yang. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359, 2010.

[72] J. Peters and S. Schaal. Natural Actor-Critic. *Neurocomputing*, 71(7-9):1180–1190, 2008.

[73] R. Pfeifer and C. Scheier. *Understanding Intelligence*. The MIT Press, Cambridge, Massachusetts, 1999.

[74] P. Poupart and N. A. Vlassis. Model-based Bayesian Reinforcement Learning in Partially Observable Domains. In *ISAIM*, 2008.

[75] L. D. Pyeatt and A. E. Howe. Decision Tree Function Approximation in Reinforcement Learning. Technical report, In Proceedings of the Third International Symposium on Adaptive Systems: Evolutionary Computation and Probabilistic Graphical Models, 1998.

[76] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-taught Learning: Transfer Learning from Unlabeled Data. In *ICML*, pages 759–766, 2007.

[77] L. Rendell, L. Fogarty, W. Hoppitt, T. Morgan, M. Webster, and K. Laland. Cognitive culture: theoretical and empirical insights into social learning strategies. *Trends in Cognitive Science*, 15(2):68–76, 2011.

[78] M. Riedmiller. Neural Fitted Q Iteration - First Experiences with a Data Efficient Neural Reinforcement Learning Method. In *ECML*, pages 317–328, 2005.

[79] M. Riedmiller. Learning Machines that Perceive, Act, and Communicate. In *Proceedings of the Second Workshop on Machine Learning for Interactive Systems (MLIS)*. ACM ICPS, 2013.

[80] L. D. Riek. Wizard of oz studies in hri: A systematic review and new reporting guidelines. *Journal of Human-Robot Interaction*, 1(1):199–136, 2012.

[81] B. Rosman and S. Ramamoorthy. A Multitask Representation Using Reusable Local Policy Templates. 2012.

[82] M. Salem, S. Kopp, I. Wachsmuth, K. Rohlfing, and F. Joublin. Generation and evaluation of communicative robot gesture. *International Journal of Social Robotics*, 4(2):201–217, 2012.

[83] J. Schatzmann, K. Weilhammer, M. N. Stuttle, and S. Young. A Survey of Statistical User Simulation Techniques for Reinforcement-Learning of Dialogue Management Strategies. *Knowledge Eng. Review*, 21(2):97–126, 2006.

[84] B. Settles. Active Learning Literature Survey. Technical Report Technical Report 1648, University of Wisconsin, Madison, 2009.

[85] Ö. Simsek, A. P. Wolfe, and A. G. Barto. Identifying Useful Subgoals in Reinforcement Learning by Local Graph Partitioning. In *ICML*, pages 816–823, 2005.

[86] M. Snel and S. Whiteson. Multi-task reinforcement learning: shaping and feature selection. In *Recent Advances in Reinforcement Learning*, pages 237–248. Springer, 2012.

[87] P. Stone and M. M. Veloso. Multiagent Systems: A Survey from a Machine Learning Perspective. *Autonomous Robots*, 8(3):345–383, 2000.

[88] H. Suay and S. Chernova. Effect of human guidance and state space size on interactive reinforcement learning. In *RO-MAN, 2011 IEEE*, pages 1–6, 2011.

[89] R. S. Sutton and A. G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998.

[90] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *NIPS*, pages 1057–1063, 1999.

[91] R. S. Sutton, J. Modayil, M. Delp, T. Degris, P. M. Pilarski, A. White, and D. Precup. Horde: a scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In L. Sonenberg, P. Stone, K. Tumer, and P. Yolum, editors, *AAMAS*, pages 761–768. IFAAMAS, 2011.

[92] R. S. Sutton, D. Precup, and S. Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1-2):181–211, 1999.

[93] C. Szepesvári. *Algorithms for Reinforcement Learning*. Morgan and Claypool Publishers, 2010.

[94] I. Szita. Reinforcement learning in games. In M. Wiering and M. van Otterlo, editors, *Reinforcement Learning: State-of-the-Art*, chapter 17. Springer, 2012.

[95] M. Tan. Multi-Agent Reinforcement Learning: Independent versus Cooperative Agents. In *ICML*, pages 330–337, 1993.

[96] M. Taylor and P. Stone. Transfer Learning for Reinforcement Learning Domains: A Survey. *JMLR*, 10:1633–1685, 2009.

[97] M. E. Taylor and P. Stone. Cross-domain transfer for reinforcement learning. In *Proceedings of the Twenty Fourth International Conference on Machine Learning (ICML 2007)*, Corvallis, Oregon, June 2007.

[98] G. Tesauro. Temporal Difference Learning and TD-Gammon. *Commun. ACM*, 38(3):58–68, 1995.

[99] A. L. Thomaz and C. Breazeal. Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence*, 172:716 – 737, 2008.

[100] S. Thrun. *Learning To Learn: Introduction*. Kluwer Academic Publishers, 1996.

[101] S. Thrun and J. O'Sullivan. Discovering Structure in Multiple Learning Tasks: The TC Algorithm. In *ICML*, pages 489–497, 1996.

[102] R. Toris, H. B. Suay, and S. Chernova. A practical comparison of three robot learning from demonstration algorithms. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, HRI '12, pages 261–262, New York, NY, USA, 2012. ACM.

[103] L. Torrey, J. Shavlik, T. Walker, and R. Maclin. Skill acquisition via transfer learning and advice taking. In *Proceedings of the Seventeenth European Conference on Machine Learning (ECML'06)*, pages 425–436, Berlin, Germany, Sept. 2006.

[104] W. Uther and M. Veloso. Adversarial Reinforcement Learning. 1997.

[105] M. Valko, M. Ghavamzadeh, and A. Lazaric. Semi-Supervised Apprenticeship Learning. *JMLR: EWRL10 Workshop and Conference Proceedings*, 24:131–141, 2012.

[106] M. van Otterlo. *The Logic of Adaptive Behavior: Knowledge Representation and Algorithms for Adaptive Sequential Decision Making under Uncertainty in First-Order and Relational Domains*. IOS Press, Amsterdam, The Netherlands, 2009.

[107] M. van Otterlo. Solving relational and first-order logical markov decision processes: A survey. In M. Wiering and M. van Otterlo, editors, *Reinforcement Learning: State-of-the-Art*, chapter 8. Springer, 2012.

[108] U. von Luxburg. A Tutorial on Spectral Clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[109] M. Waibel, M. Beetz, J. Civera, R. d'Andrea, J. Elfring, D. Galvez-Lopez, K. Haussermann, R. J. M. Janssen, J. M. M. Montiel, A. Perzylo, B. Schiessle, M. Tenorth, O. Zweigle, and M. J. G. van de Molengraft. Roboearth – a world wide web for robots. *IEEE Robotics & Automation Magazine*, 18(2):69–82, 2011.

[110] M. Wiering and M. van Otterlo. *Reinforcement Learning: State-of-the-Art*. Springer, 2012.

[111] H. Xiong, S. Szedmak, and J. Piater. Homogeneity Analysis for Object-Action Relation Reasoning in Kitchen Scenario. In *Proceedings of the Second Workshop on Machine Learning for Interactive Systems (MLIS)*. ACM ICPS, 2013.

[112] D. Yarowsky. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *ACL*, pages 189–196, 1995.

[113] J. Young, J. Sung, A. Voida, E. Sharlin, T. Igarashi, H. Christensen, and R. Grinter. Evaluating human-robot interaction. *International Journal of Social Robotics*, 3(1):53–67, 2011.

[114] S. Young, M. Gasic, B. Thomson, and J. D. Williams. POMDP-Based Statistical Spoken Dialog Systems: A Review. *Proceedings of the IEEE*, 101(5):1160–1179, 2013.

[115] S. Zhifei and E. M. Joo. A survey of inverse reinforcement learning techniques. *International Journal of Intelligent Computing and Cybernetics*, 5(3):293–311, 2012.

[116] X. Zhu. Semi-Supervised Learning Literature Survey. Technical Report Technical Report 1530, University of Wisconsin, Madison, 2006.