Beyond Clever Hans: Learning From People Without Their Really Trying

by

Vivek Veeriah Jeya Veeraiah

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

University of Alberta

Abstract

Facial expressions and other body language are important for human communication. They complement speech and make the process of communication simple and sustainable. However, the process of communication using existing approaches to human-machine interaction is not intuitive as that of human communication. Specifically, the existing approaches to human machine interaction do not learn from whatever subtle non-verbal cues produced by a user. Many of the existing approaches map body language cues to instructions or rewards and use them to train a machine. These mappings are not learned from ongoing interactions and are assumed to be defined by some external source. As a consequence, the communicative process through these approaches can cause significant cognitive load on the user. This is an important problem that needs to be addressed if we are to amplify our existing cognitive and physical capabilities through intelligent machines. Towards addressing this, we introduce our idea that allows machines to learn from whatever subtle cues people produce during the process of interaction. Particularly, the agent learns a value function that maps the user's non-verbal cues to later rewards. By using this value function, the user can teach their agent to complete a task according to their preferences, and consequently maximize their satisfaction. We demonstrate this by training an agent using facial expressions. Furthermore, we show that these learned value functions can be successfully transferred across tasks. In conclusion, our approach is the first that allows people to teach their machines using whatever subtle cues they produce and could take us far in achieving sustainable forms of human-machine interaction.

To Mom and Dad

Acknowledgements

I would like to thank my supervisors, Richard S. Sutton and Patrick M. Pilarski, for being amazing mentors and for introducing me to this exciting field of reinforcement learning. In particular, I am thankful to Rich for showing me how to critically develop my own thoughts, and more importantly, for inspiring me to pursue ambitious and important ideas. I would like to thank Patrick for his never-ending enthusiasm towards research and for his consistent encouragement. His excitement has always been contagious!

Further, I would like to thank Pooria Joulani, Craig Sherstan, Rupam Mahmood, Roshan Shariff, Marlos Machado, Kenny Young, Tian Tian, Eric Graves, Gautham Vasan and Sanket Kumar Singh for all their help along the way. My experiences at University of Alberta and at the RLAI Lab wouldn't have been a joyful one without them.

Table of Contents

List of Figures

Chapter 1

Beyond Clever Hans

The overarching goal of this thesis is to introduce an approach to human-machine interaction, where the machine learns from whatever subtle non-verbal cues that are produced by the user during the process of interaction. We begin by introducing the story of Clever Hans, which is a popular example that illustrates the role of body language in communication. Subsequently, we broadly discuss how existing approaches use body language for human-machine interaction. Finally, we introduce our approach and highlight our contributions in this thesis.

1.1 Horse that Answered Questions

Wilhelm von Osten was a math teacher and an amateur horse trainer. He owned a horse called Hans. In the year 1891, von Osten displayed his horse in public and claimed that he had taught his horse to perform simple arithmetic calculations, tell time, and keep track of the calendar. Hans answered many of the questions posed by the questioner, and over time, became a public sensation in the late twentieth century. As a result of this popularity, the horse was called "Clever Hans" and was reported by the New York Times in 1904.

The significant public interest in Clever Hans motivated the German board of education to appoint a commission to investigate whether the horse possessed intelligence. The commission recruited a German psychologist named Oskar Pfungst, who conducted a substantial number of trials with Hans and

discovered that Hans got the right answer only when the questioner knew what the answer was. More importantly, Hans needed to see the questioner while answering their questions.

Pfungst then studied the behavior of the questioner while the horse was answering their question and reported his findings (c.f. Pfungst, 1911). He discovered that the horse's behavior was influenced by subtle and unintentional cues. Specifically, as Hans approached the correct answer the questioner's posture and facial expressions changed in ways that were consistent with an increase in tension, which was released when Hans made the final tap with its hoof. This change in body language served as a useful cue for the horse to stop tapping its hoof.

Hans learned when to stop tapping its hoof by observing the subtle non-verbal cues of the questioner. This ability of the horse was wrongly attributed to be a consequence of possessing intelligence. While training his horse, von Osten rewarded the horse whenever it answered a question correctly which reinforced this process of answering from the questioner's subtle cues.

1.2 Human-Machine Interaction using Body Language

The ideal approaches to human-machine interaction should allow users to communicate naturally as they would do with their peers. This would make the process of interaction between people and machines to be sustainable and scalable. As an important step towards achieving this ambitious goal of ideal human-machine interaction, we need to design approaches that allow machines to learn from whatever non-verbal cues produced during the interaction, similar to how the horse picked on unintentional non-verbal cues of its questioner for answering their questions.

In human communication, the subtle non-verbal cues that we produce unintentionally influences the observer's behavior towards us. This is a natural process and is well-studied in psychology (De Gelder, 2006; Pezzulo et al., 2013; Jack & Schyns, 2015) and in affective neuroscience (Adolphs, 2002; Whalen et

al., 2013). Specifically, these body language cues have communicative value to their observers and allowing the machines to learn from our subtle non-verbal cues seems to be a natural direction to pursue in human-machine interaction.

The existing approaches to human-machine interaction do not use body language as training information. Specifically, these approaches assume that the body language cues and their meaning are given to the machine from some external source, instead of learning them from ongoing interactions. The existing approaches use non-verbal cues for either instructing or rewarding the machines. Particularly, the system designer maps body language cues to instructions which the machine performs on observing the corresponding cue from the user. Some approaches also map body language to rewards which are provided to the machine.

Using body language as instructions or rewards requires the user to translate their non-verbal cues to effectively teach the machine according to their preferences. The approaches that use body language as instructions or rewards assume that people produce similar body language during interaction. This is not a reasonable assumption to make because body language differs significantly with people, usually depending on their situations and cultural norms (Roselli & Ardila, 2003; Yammiyavar et al., 2008). Requiring the user to actively translate their cues in order to meet the machine's design can cause significant cognitive load on the user, especially in real-world human-machine interaction tasks (Hollender et al., 2010; Fridman et al., 2017; Mathewson & Pilarski, 2017). As a consequence of these reasons, the existing approaches are unsustainable and leads to unsuccessful human-machine partnerships.

1.3 Moving Beyond Clever Hans

In this thesis, we introduce a human-machine interaction approach that is inspired from Clever Hans and investigate whether this brings us closer towards natural human-machine collaboration.

In human-machine interaction domains, people naturally produce subtle non-verbal cues to indicate their satisfaction towards their machines. In the

approach that we introduce here, the machine learns to associate their users' subtle cues with their satisfaction and subsequently uses this to learn an appropriate behavior in a given task. More specifically, the machine learns an evaluation function that predicts its user's satisfaction from their non-verbal cues. By using these predictions, the machine learns to adapt its behavior in a given task according to its user's preferences and, as a result maximize user satisfaction. Here, we view the machine's performance in a task as a measure of user satisfaction.

We formalize and implement our approach using reinforcement learning. Particularly, from reinforcement learning, we use techniques to learn value functions. In our approach, the machine learns a *value function* that associates user's non-verbal cues with their satisfaction, which is measured in terms of their occasional explicit feedback. The machine predicts its user's satisfaction using this value function, and as a result the machine perceives a feedback signal in the form of body language cues. This evaluative signal drives the machine's behavior within a given task according to their user's preferences, thereby maximizing their satisfaction.

As opposed to existing approaches that use body language in human-machine interaction, our approach does not require the system designer to define their meaning to the machine. Specifically, the machine learns the meaning of the various non-verbal cues directly form the ongoing interactions with its user. Another important distinction of our approach is that the machine learns from ongoing interactions to predict user's satisfaction directly from their body language and does not require a supervised training dataset.

This is more general than mimicking Clever Hans to a human-machine interaction setting, because Hans only learned when to stop tapping its hoof from subtle cues as opposed to learning how much the user was satisfied. More importantly, by learning to predict user satisfaction, our approach allows the machine to perform tasks that are truly important to the user.

As our approach relies on whatever subtle cues produced during the process of interaction, it does not require the user to translate or exaggerate their non-verbal cues. The approach learns from whatever subtle non-verbal cues

produced by the user while interacting with their machine. We believe this to be the first work in directly addressing the overall goal of this thesis, where an implemented system learns from people's body language, without involving much effort.

1.4 Outline

Here, we give a brief outline for the rest of this thesis:

Chapter 2 introduces a few concepts of reinforcement learning that are used in Chapters 4 and 5.

Chapter 3 presents a background on existing methods used for human-machine interaction.

Chapter 4 introduces our *prospective body language* approach which allows the users to teach their machines using whatever subtle body language cues that are produced during the process of interaction. This chapter also grounds our contribution with the existing approaches that allow users to teach their machines using body language. This chapter is the fundamental contribution of this thesis.

Finally, in Chapter 5 and 6, we present experimental results of our approach of learning from prospective body language that is implemented using two popular reinforcement learning techniques. The results show that our approach can improve over a conventional user-interaction agent in difficult and changing tasks.

Lastly, in Chapter 7, we discuss few common concerns related to our approach and present future directions of research.

1.5 Contributions

The key contributions of this thesis are summarized as follows:

- We introduce our interactive machine learning approach that allows people to teach machines using subtle body language cues, without involving

much effort. We view this as the initial steps towards natural and intuitive human-machine collaboration.

- We further develop our approach and implement it using two different reinforcement learning techniques and empirically evaluate them on a simulated grip-selection domain.

- We discuss extensions to our approach as potential future directions towards achieving simple human-machine interactions.

Chapter 2

Background

In this chapter, we introduce and define the different concepts from reinforcement learning, along with the necessary algorithms, that are required for understanding this thesis.

2.1 The Reinforcement Learning Problem

The reinforcement learning problem involves an agent that continually interacts with its environment in order to achieve a certain goal, where this goal is defined in terms of a reward signal.

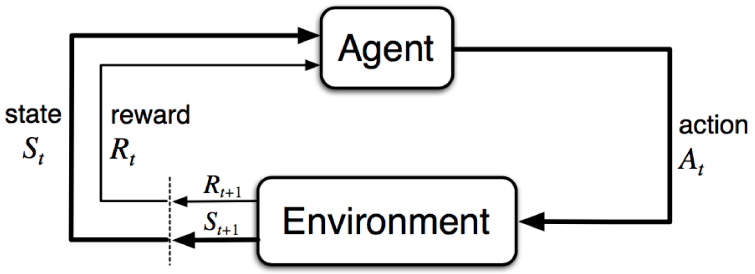


Figure 2.1: The Reinforcement Learning problem: Interaction between an agent and its environment

A simple representation of the RL problem is shown in Figure 2.1. At each time step, the agent observes its current state, then chooses an action that influences the environment. In response, the environment provides a reward and a next state to the agent. This process repeats continually where the agent interacts with its environment. The objective for the agent is to learn and select actions so as to maximize this numerical reward signal.

A reinforcement learning problem is represented as a Markov Decision Process (MDP) with a finite set $\mathcal{S}$ of N states and a finite set $\mathcal{A}$ of actions, with a discount factor of $\gamma \in [0, 1]$. At each time step t, the agent observes its current state $S_t \in \mathcal{S}$ where $\mathcal{S}$ is a set of all possible states and $|\mathcal{S}| = N$ is the total number of states available in the environment. After observing its current state S_t, the agent chooses an action $A_t \in \mathcal{A}$. This action influences the environment in producing a scalar reward $R_{t+1} \in \mathcal{R}$. This reward signal R_{t+1} is generally a function of the agent's current state S_t and action A_t. Following this reward, the agent transitions into the next state $S_{t+1} \in \mathcal{S}$ and this entire interactive process repeats continually.

The expected value of this reward signal $r(s, a)$ is defined as the the average value of rewards observed at a particular state s after taking a particular action a:

$$r(s, a) = \mathbb{E}\Big[R_{t+1}|S_t = s, A_t = a\Big]$$

The environment, after receiving an action a from the agent, transitions to the next state s' from the current state s where this transition is defined by a probability function $p(s'|s, a)$. Specifically, this transition probability function gives the likelihood of the agent transitioning into a state s' from the current state s, after picking an action a.

$$p(s'|s, a) = \Pr\Big\{S_{t+1} = s'|S_t = s, A_t = a\Big\}$$

It is important to note here that the transition probability $p(s'|s, a)$ and the reward function $r(s, a)$ are specific to an environment and are unknown to the learning agent.

The goal for a reinforcement learning agent is to maximize the scalar rewards it receives over time, through ongoing interactions in the form of state observations and actions with the environment. In order to achieve this, the agent needs to learn to pick actions that eventually produce higher rewards and map these actions with their current situations (i.e. the states). This association of actions to states is formally called as *policy* and is denoted as

$\pi(a|s)$. The policy gives the likelihood of picking an action a on observing a state s.

Finally, $\gamma \in [0, 1]$ is a discount parameter representing the relative importance of future rewards compared to the importance of the immediate reward. The goal for the agent is to maximize this discounted sum of rewards that it receives over time by learning an action-selection policy π. This discounted sum of rewards is called as *return*, which is formally defined as:

$$G_t = \sum_{i=0}^{\infty} \gamma^i R_{t+i+1}$$

When γ is closer to 0, the return is computed by assigning a higher weight to immediate rewards rather than those that are received later in the future. In other words, when the discount factor is close to 0, the agent learns a *myopic* policy that maximizes immediate rewards. Setting a discount factor closer to 1 allows the agent to learn a policy that maximizes future rewards.

2.2 Value Functions

A value function is the fundamental idea in reinforcement learning (Sutton & Barto, 2017). The central idea behind a value function is to cache the utility or knowledge of a state s in a single function $v_\pi(s)$, which can be used by the agent in achieving its overall goal.

The general approach for solving a reinforcement learning problem involves learning this value function. This is achieved through a certain class of algorithms called Temporal-Difference (TD) learning. At its core, TD incrementally learns a value estimate of a state by *bootstrapping* from its succeeding states' estimates.

A value function is defined as an estimate of the discounted sum of rewards that would be achieved by the agent by following the current action-selection policy π. Informally, a value function tells the agent how "good" its action-selection policy is. For a given policy π, we can now define the state-value function $v_\pi(s)$ as a function that maps a state s to its expected return, that

would be obtained starting from the state s and following the policy π. Mathematically, a value function is defined as:

$$v_\pi(s) = \mathbb{E}_\pi\Big[\sum_{i=0}^{\infty} \gamma^i R_{t+i+1} | S_t = s\Big] \tag{2.1}$$

The state-value function can be written in a recursive form based on the value of the subsequent states:

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_\pi\Big[\sum_{i=0}^{\infty} \gamma^i R_{t+i+1} | S_t = s\Big] \\ &= \mathbb{E}_\pi\Big[R_{t+1} + \gamma \sum_{i=1}^{\infty} \gamma^i R_{t+i+1} | S_t = s\Big] \\ &= \sum_{a\in\mathcal{A}} \pi(a|s) \sum_{s'\in\mathcal{S}} \sum_{r\in\mathcal{R}} p(s', r|s, a)\Big[r + \gamma \mathbb{E}_\pi\big[\sum_{i=1}^{\infty} \gamma^i R_{t+i+1} | S_{t+1} = s'\big]\Big] \\ v_\pi(s) &= \sum_{a\in\mathcal{A}} \pi(a|s) \sum_{s'\in\mathcal{S}} \sum_{r\in\mathcal{R}} p(s', r|s, a)\Big[r + \gamma v_\pi(s')\Big] \end{aligned} \tag{2.2}$$

This is called a state-value function because it does not take into account of the action a that is picked from by an agent using the policy π at every time step. Although, the return that can be obtained by an agent depends on the action a that it chooses for a given state s, this value function marginalizes the effect of picking this action for the given state. Specifically, for this reason, it is called the state-value function.

An obvious extension of this would be the state-action value function or the action-value function. Informally, the action-value function $q_\pi(s, a)$ maps a state-action pair *(s, a)* to the expected return starting from state s, taking the action a and following the policy π thereon:

$$q_\pi(s, a) = \mathbb{E}_\pi\Big[\sum_{i=0}^{\infty} \gamma^i R_{t+i+1} | S_t = s; A_t = a\Big]$$

This can be also be written as a recursive equation based on the action-value function of subsequent states from the current state s:

$$q_\pi(s, a) = \sum_{s'\in\mathcal{S}} \sum_{r\in\mathcal{R}} p(s', r|s, a)\Big[r + \gamma q_\pi(s', a')\Big] \tag{2.3}$$

2.3 Temporal-Difference (TD) Learning

As previously described, one of the central ideas in reinforcement learning is the estimation of value function. The learning agent needs to estimate the value of a particular state or state-action pair in order to improve its policy π. The value function estimates the sum of discounted sum of future rewards, as shown in Equation 2.1. More importantly, TD forms the fundamental basis for many prediction and control algorithms applied in reinforcement learning.

Sutton (1988) introduced a class of algorithms for learning these value functions, called *temporal-difference* (TD) learning algorithms. These TD algorithms are a significant contribution to the field of reinforcement learning as they allow in *incrementally* learning a value function, making it suitable for online learning. Also, these TD learning algorithms support the use *function approximation* and *bootstrapping*.

Function approximation is a technique for representing a state when there are uncountably many states using a tuneable parameterized function. Generally, the number of parameters of this function approximator is negligible compared to the number of states present in an environment.

The simplest approach for estimating a value function for many states is to create a look-up table with one entry per state and update them based on the Equation 2.2. For learning a action-value function, the updates are based on Equation 2.3. However, it is difficult to represent all possible states (or state-action pairs) as a look-up table in many realistic settings. In order handle such large domains, this technique of function approximation was introduced.

Compressing the space of all states into a set of tractable and computationally efficient representation of the state provides an additional advantage of generalizing across similar states, which can accelerate learning. Some well known examples of function approximation in reinforcement learning are linear function approximators, kernels and neural networks.

Bootstrapping refers to a term used in TD learning and has the same meaning as in dynamic programming. Bootstrapping is an efficient way of updating estimates from other predictions. Usually, this means that we can

make updates to estimates of states before its actual outcome is observed. However, this can skew the learned estimates, thereby introducing bias.

Without bootstrapping, the estimates are updated based on the actual outcomes that are observed. Such algorithms are called Monte Carlo methods and occupy a spectrum on the opposite side of TD algorithms. Monte Carlo methods make learning updates only at the end of an episode, when the final state is reached. These methods suffer from high variance in their estimates, especially in stochastic domains as the returns vary a lot. On the other hand, TD algorithms can make incremental learning updates at each time step through bootstrapping. As a direct consequence of bootstrapping, TD algorithms have less variance in their estimates.

Algorithm 1 TD(λ) Learning Algorithm

INPUT: $\alpha, \lambda, \boldsymbol{w}_{init}$
  $\boldsymbol{w} \leftarrow \boldsymbol{w}_{init}$ ▷ $\boldsymbol{w}$ is the weight vector for the state-value function
  $\boldsymbol{e}_w \leftarrow 0$ ▷ $\boldsymbol{e_w}$ is the eligibility trace for the state-value function
  for num. of episodes **do**
    obtain initial state S
    $\phi \leftarrow$ feature corresponding to S
    while S is not terminal **do**
      obtain next state S' and reward R
      $\phi' \leftarrow$ feature corresponding to S'
      $\delta \leftarrow R + \gamma \boldsymbol{w}^\top \phi' - \boldsymbol{w}^\top \phi$
      $\boldsymbol{e}_w \leftarrow \gamma\lambda \boldsymbol{e}_w + \phi$
      $\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha\delta \boldsymbol{e}_w$
      $S \leftarrow S'$
      $\phi \leftarrow \phi'$
    end while
  end for

TD algorithms are formulated as *forward view* and *backward view* methods. The forward view of TD looks many time steps into the future, then computes and makes incremental learning updates to its weight vectors. This forward view of TD cannot be naïvely implemented as an online learning algorithm because it relies on information extending many time steps into the future. It was introduced as a theoretical framework for studying and designing TD algorithms (Sutton, Mahmood & White, 2016; van Seijen et al., 2016; van

Seijen, 2016). The backward view of TD uses information at the current time step and makes incremental updates to the weight vector. More importantly, these backward view of TD can be implemented for online settings. In expectation, the predictions made by the backward and forward views of TD are equal.

$$\boldsymbol{e}_t = \gamma\lambda\boldsymbol{e}_{t-1} + \phi_{t-1} \tag{2.4}$$

$$\boldsymbol{e}_t = \max(\gamma\lambda\boldsymbol{e}_{t-1}, \phi_{t-1}) \tag{2.5}$$

For computational feasibility, the backward view of TD includes an eligibility trace vector $\boldsymbol{e} \in \mathcal{R}^n$, such as the accumulating traces (Equation 2.4) or replacing traces (Equation 2.5).

2.3.1 Prediction Algorithm: TD(λ) Learning

TD(λ) algorithm is the most effective prediction algorithm in reinforcement learning. The algorithm predicts the return G_t achieved by an agent by following a particular action-selection policy and was introduced in Sutton (1988). It is summarized in Algorithm 1.

This algorithm uses a parameter $\lambda \in [0, 1]$ for trading-off between bias and variance. Specifically, by selecting a value for λ, the resulting TD algorithm lies somewhere between a full bootstrapping method ($\lambda = 0$) and a Monte Carlo method ($\lambda = 1$). The learning performance of an agent making predictions using TD depends on this tuneable parameter and is domain-specific.

The goal of the algorithm is to learn a set of weights $\boldsymbol{w} \in \mathcal{R}^n$ such that the value can be estimated accurately for a given state $s \in \mathcal{S}$, where this state is represented by its feature vector $\phi(s) \in \mathcal{R}^n$. The estimated value of this state is given via a simple dot product between the weights and the features of a state:

$$\hat{v}(S, \boldsymbol{w}) = \boldsymbol{w}^\top \phi(S)$$

The representation of the state $s \in \mathcal{S}$ is represented by a simple feature vector $\phi(s) \in \mathcal{R}^n$. One could also represent the state as a nonlinear function. However, linear function approximation is computationally efficient and is sufficient for understanding the rest of this thesis.

The term δ in Algorithm 1 is called the Temporal-Difference (TD) error. It is defined as the difference between the bootstrapped target estimate (i.e., $R_{t+1} + \gamma \hat{v}(S_{t+1}, \boldsymbol{w})$) and the current estimate of the state. When TD makes a perfect prediction for a given state then this TD error will be zero.

The weight update rule for the TD algorithm in Algorithm 1 is derived from the standard stochastic gradient descent algorithm. The term α is a step-size parameter that controls the amount to move in the direction of gradient at a given time step. This step-size parameter is subject to certain constraints, so that the algorithm can converge to a local solution. In practice, this α parameter is set to a small constant value, allowing the agent to learn and adapt to non-stationary (i.e., changing) environments.

2.3.2 Control Algorithm: Sarsa(λ) algorithm

TD learning methods are useful for making predictions about the future rewards received by the agent interacting with an environment for a given action-selection policy. An interesting extension of TD algorithm is for learning the state-action values, which implicitly represents an action-selection policy π that maximizes the rewards achieved by the learning agent from its interactions. The algorithms that learn how to select actions are referred to as control algorithms and one such algorithm is summarized in Algorithm 2.

An action-value method learns a policy π that maps states with actions. This policy is represented implicitly through action-values corresponding to state-action pairs. Sarsa is a popular TD control algorithm that starts out by evaluating the utility of following its current action-selection policy. While choosing an action, the agent picks the action that has a large state-action estimate and this action is called a greedy action. The agent could also randomly pick a non-greedy action with a certain probability ϵ, and such a policy is called ϵ-greedy policy. Specifically, the agent selects a greedy action with a

Algorithm 2 Sarsa(λ) Learning Algorithm with Accumulating Traces

INPUT: $\alpha, \lambda, \gamma, \boldsymbol{w}_{init}$
  $\boldsymbol{w} \leftarrow \boldsymbol{w}_{init}$ ▷ $\boldsymbol{w}$ is the weight vector for the action-value function
  $\boldsymbol{e}_w \leftarrow 0$ ▷ $\boldsymbol{e}_{\boldsymbol{w}}$ is the eligibility trace for the action-value function
  for num. of episodes **do**
    obtain initial state S
    select action A based on state S (for example, ϵ-greedy)
    $\phi \leftarrow$ features corresponding to S, A
    while S is not terminal **do**
      take action A, observe next state S' and reward R
      select action A' based on state S'
      $\phi' \leftarrow$ features corresponding to S', A'
      $\delta \leftarrow R + \gamma \boldsymbol{w}^\top \phi' - \boldsymbol{w}^\top \phi$
      $\boldsymbol{e}_w \leftarrow \gamma \lambda \boldsymbol{e}_w + \phi$
      $\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha \delta \boldsymbol{e}_w$
      $S \leftarrow S'$
      $A \leftarrow A'$
      $\phi \leftarrow \phi'$
    end while
  end for

probability of $1 - \epsilon$ and a non-greedy action with a probability of ϵ.

This ϵ parameter can take a value between 0 and 1, and ensures that the learning agent occasionally explores by taking a non-greedy action. In non-stationary environments, the action that results in higher rewards slowly drifts over time and using an ϵ-greedy can lead the agent to achieve better rewards in these cases.

2.4 Actor-Critic Methods

In reinforcement learning, sometimes, it is better to decouple the policy from value function and learn them separately from ongoing interactions, rather than learning it indirectly via state-action values. Actor-Critic methods are those methods that meet these requirements, learning an action-selection policy from online experiences. An example actor-critic architecture is shown in Figure 2.2.

The simplest approach for learning a control policy is by directly extending the value function based method, like TD learning, to learn state-action

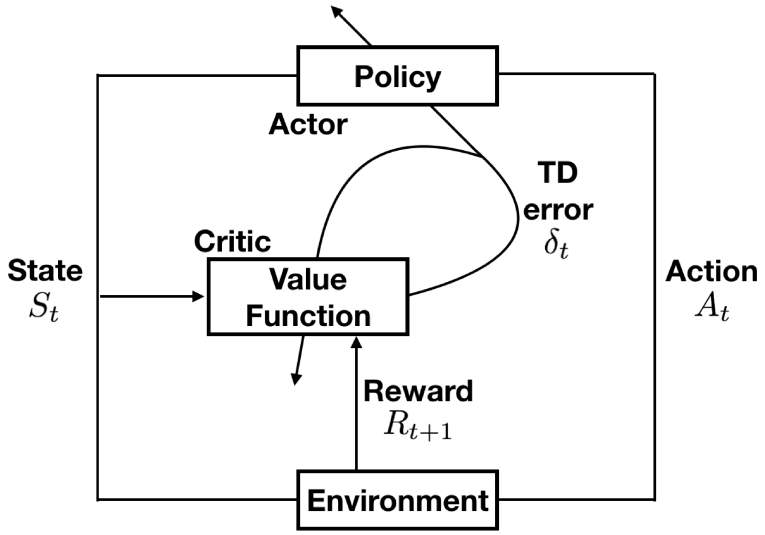


Figure 2.2: The Actor-Critic architecture

values. This results in effective control algorithms (namely, Sarsa and Q-learning). Though, these control algorithms are sufficient for domains with small number of actions, they actually result in poor learning performance in domains with many actions. More importantly, in many robotic domains, it is important to learn an action-selection policy that chooses real-valued actions. For example, in order to control a servo motors, the agent needs to use real-values signals. As a consequence of addressing these concerns, actor-critic methods were introduced (Barto et al., 1983; Konda & Tsitsiklis, 2000; Bhatnagar et al., 2009).

Actor-Critic methods are control methods that learn an explicit policy through TD principles. It consists of two seperable modules: actor, which learns an action-selection policy and a critic, that critiques the actions selected by the actor module. Specifically, instead of learning values for each state-action pairs, actor-critic methods learn to directly learn a function, called the actor, that maps states with actions. The critic is responsible for generating TD-errors, critiquing the actions selected by the actor module. This TD error drives learning in both actor and critic modules. These modules can be parameterized separately with function approximators. The overall algorithm of actor-critic method is summarized in Algorithm 5.

Algorithm 3 Actor-Critic with Accumulating Traces

INPUT: $\alpha_v, \alpha_\pi, \lambda, \gamma, \boldsymbol{w}_{init}, \boldsymbol{\theta}_{init}$

```
w ← w_init                          ▷ w is the weight vector for the critic
θ ← θ_init                          ▷ θ is the weight vector for the actor
e_w ← 0                             ▷ e_w is the eligibility trace for the critic
e_θ ← 0                             ▷ e_θ is the eligibility trace for the actor
for num. of episodes do
    obtain initial state S and φ
    while S is not terminal do
        select action A for state S (using φ)
        take action A, observe S' (along with φ') and R
        δ ← R + γ w^⊤ φ' − w^⊤ φ
        e_w ← γλ e_w + φ
        w ← w + α_v δ e_w
        e_θ ← γλ e_θ + ∇_θ log[π(A|S, θ)]
        θ ← θ + α_π δ e_θ
        S ← S'
        φ ← φ'
    end while
end for
```

2.5 Linear Function Approximation Using Tile Coding

Tile coding is a popular and simple nonlinear approach for generating feature vector $\boldsymbol{\phi}(\boldsymbol{s}) \in \mathcal{R}^n$. The approach generates a sparse, binary feature vector from real-valued signals obtained as the state from the environment. Because tile coding produces a sparse feature vector from real-valued signals, it is well-suited for online learning in reinforcement learning. It has produced many successful robotic applications.

Tile coding uses a *tiling* over the real-valued state space, partitioning it into non-overlapping regions called *tiles* as shown in Figure 2.3. The Figure 2.3 shows a two-dimensional tiling for hypothetical real-valued state space. The tiles in this tiling do not have to be in the shape of a square and need not have the same resolution for different regions of the state space. Furthermore, they can be used with any number of dimensions in the state space.

For a given tiling, the real-valued signal of the state would be active (i.e., present) in exactly one of the tile whereas all the other tiles are inactive. By

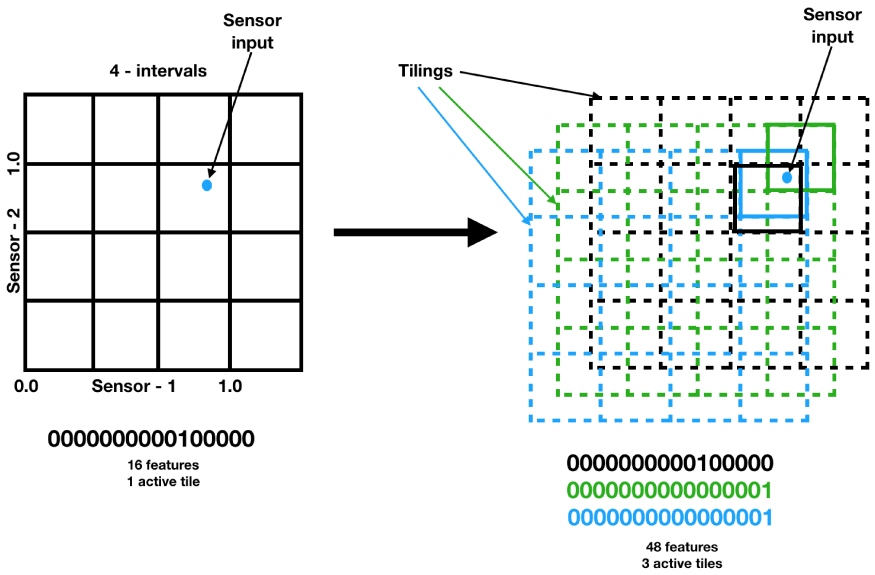


Figure 2.3: Tile coding

encoding this presence of the state signal through this tiling produces a binary vector of 0s corresponding to those inactive tiles and a 1 in the position of an active tile. The same process can be repeated for a number of tilings that are offset by predetermined or a random pattern from each other, each of them producing a single binary vector. All of these binary vectors are concatenated to form a long and sparse binary vector, which forms the feature representation $\boldsymbol{\phi(s)}$ for a given state signal $s \in \mathcal{S}$.

Chapter 3

Existing Approaches to Human-Machine Interaction

Designing intelligent, user-interactive agents, that can effectively amplify our existing cognitive and physical capabilities, are one of the important promises of Artificial Intelligence. Through this, we hope to merge ourselves with AI. A principle example of this can be found in assistive rehabilitation robots, where machine learning techniques enable electromechanical systems in restoring or augmenting biological limbs that are lost through injury or illness.

Interactive machine learning and human-computer interaction are fields that are as old as computers and it is not possible to summarize all of the key ideas in this thesis. However, we summarize some of the recent approaches in these fields, particularly those that are related to this thesis, followed by their strengths and weaknesses.

The existing works in interactive machine learning approach the problem from multiple perspectives. However, most of them share a common goal of allowing people to effectively communicate their intents or preferences to a machine, which then uses this information to solve a given task.

3.1 Interactions in the Form of Rewards

The simplest approach for empowering people to communicate their preferences to an interacting machine is by providing user-generated rewards. This user-generated reward augments or shapes the existing reward signal produced

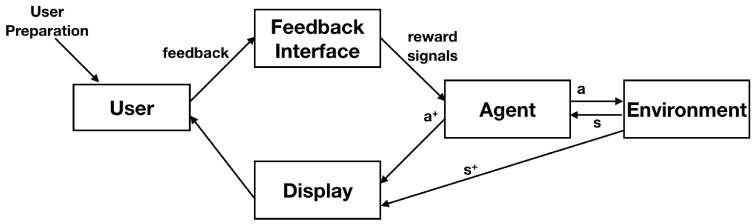


Figure 3.1: Interactive Shaping: Providing interactions in the form of rewards to the learning agent.

by an environment within which the agent is located. The primary motivation behind this line of research is from animal training, where a trainer uses a clicker and some treats to shape the behavior of an animal. In this research direction, the user directly influences the reward channel of the learning system through *explicitly* generated human reward. A simple representation of this line of research is shown in Figure 3.1.

The first research work that explored this idea of shaping a learning system's behavior by influencing the reward channel is the one introduced by Isbell et al. (2000). In this work, a virtual agent is located in an active online community frequented by hundreds of users. The users interacted with this agent through some predefined text based reward signals. The virtual agent, by keeping track of certain statistics, learned to adapt its behavior to maximize this user-generated reward signal.

A more recent work, conforming to this research trend, was conducted by Thomaz and Breazeal (2006, 2008). Here, the authors extensively studied how the users teach a virtual reinforcement learning agent to perform a complex task. Specifically, the users provided rewards to the learning agent with the sole intention of guiding the system to complete a given task. By analyzing the reward generation pattern produced by the users for teaching their agents, they observed that each user followed a different approach for guiding the learning system. There was no standard strategy adopted by the users. Also, it was observed that the interacting users had different views about the rewards that were provided to the agent. Particularly, some users frequently provided positive rewards with the intention of motivating the learning agent and this

vastly differs from a reinforcement learning perspective, which views rewards as evaluations for taking a particular action or a sequence of actions.

Pilarski et al. (2011) pushed this line of research forward by showing that virtual prosthetic limb agent can learn complex behaviors through online training from rewards generated by a user. Another related study, conducted by Suay and Chernova (2011), extended the work of Thomaz and Breazeal (2008) for teaching a robot to sort different objects. Both of these works were probably the first to look at teaching an interactive learning system using human-generated rewards alone.

Many of the recent works have leveraged this fundamental idea of training an interactive agent with human-generated rewards leading to many valuable contributions in the field of interactive machine learning. Knox and Stone (2013) successfully trained a robot with human-generated rewards. Iturrate et al. (2010) trained a supervised learning algorithm to classify different EEG patterns, obtained from an interacting user, as positive and negative reward signals. More recently, this approach of classifying EEG patterns as rewards, introduced by Iturrate et al. (2010), was used for teaching a neuroprosthesis arm to perform simple control tasks (Iturrate et al., 2015) and for controlling the steering angle of a car (Zhang et al., 2015).

All of these research works assumes that the distribution of human-generated rewards to be consistent throughout an experiment setting. However, this turned out to be false as per the findings of Thomaz and Breazeal (2008). Particularly because the interacting user develops a mental model of the learning agent, and as the agent improves its behavior, the frequency of user-generated reward begins to reduce. This leads the agent to unlearn, thereby degenerating its behavior towards something that is undesirable by its user.

In order to address this important issue, where an interacting user modifies his/her reward generation patterns, Knox and Stone (2009) introduced the idea of training a supervised learning algorithm with multiple state-action pairs and its corresponding user-generated reward. Once this supervised learning method reasonably learns a reward-generation model, the interacting user is replaced this reward-generator. This approach allows the interacting agent to

receive consistent rewards throughout the task.

Though this simple approach of Knox and Stone (2009) addresses an important problem in interactive machine learning, it opens up a more significantly difficult issue of adaptability in the learning agent. Specifically, when the task changes, ever so minutely, the user needs to interrupt the agent and reprogram this supervised learning method with new training data that is tailored to this modified task. Another noted disadvantage of their approach is that if the task is complex and realistic, like a prosthetic limb or a self-driving car, it is impossible scale up their approach as it requires the user to label all possible state-action pairs with a human-generated reward, which is cumbersome.

3.2 Interactions in the Form of Demonstrations and Instructions

A closely related line of research that is often used for teaching learning agents is by providing interactions in the form of demonstrations or instructions, generally provided by non-expert users. Learning from demonstration is a key idea, first introduced by Abbeel and Ng (2004), through which a learning agent can be trained to follow a behavior demonstrated by a user. This demonstration is the only interaction that happens between the user and learning agent. Specifically, an error function is formulated based on this demonstration by provided by the user, which is then used for training the agent.

Many of the subsequent research works in learning from demonstrations (e.g., Koenig and Mataric, 2012; Schulman et al., 2013; Alizadeh et al., 2014) relies on the fundamental idea described above. In order to train a learning agent to perform a particular behavior it needs to know its expected sequence of actions which is provided broadly as interactions in the form of demonstrations. Based on these demonstrations, the agent learns to take a sequence of actions that closely match these demonstrations. This limits the approach because the learning agent will not know the right action to take when faced with an unseen part of the environment. To overcome this limitation, many probabilistic approaches were developed to query the user on taking a right

sequence of actions (Cakmak & Thomaz, 2012).

Recently, Vasan and Pilarski (2017) introduced an approach for training a myoelectric prosthetic arm from online demonstrations provided by the interacting user. In their approach, they generate reward signals to the learning agent that punishes behaviors that are far from the user's demonstrations. Over time, their approach allows the agent to learn a behavior that closely matches with the interacting user.

Some research works have also looked at using explicit instructions (Breazeal, 1998; Liu & Picard, 2003), which can either be verbal or non-verbal cues, for allowing people to teach their learning agents to perform a task. Though this works in simple domains, it is unscalable for many realistic settings because it often imposes on the users' cognitive load.

3.3 Relevance of these Related Works to this Thesis

By perusing many of the research works in the field of human-machine interaction, it becomes obvious that most of the approaches are focused on shaping the agent through rewards and punishments or through demonstrations. Though, these are simple approaches towards interactive machine learning, it leads to issues in changing environments (or tasks). Specifically, whenever the task changes or modified, the user needs to provide more rewards or demonstrations in order to modify the agent's behavior.

Here, we hypothesize that if a learning agent can learn and understand the meaning of various cues from the user's body language, then the agent can quickly learn to perceive evaluative feedback from the user's body language and need not rely on explicit interactions from the user. More importantly, this allows the agent to adapt its behavior according to the user's expectations by picking on their body language cues. It is important to point out here that our approach is the first that learns the meaning of various subtle body language cues produced by the user during the process of interaction.

Chapter 4

Learning From Prospective Body Language

The fundamental contribution of this thesis is to introduce a human-machine interaction approach that allows people to teach their machines using whatever subtle body language cues that are produced during the process of interaction. As opposed to existing approaches, the meaning of these subtle cues are adaptively learned by the machine from ongoing interactions with the user. As the machine learns the meaning of these subtle cues, the user can teach the machine to perform a given task according to their preferences.

By teaching the machine through these adaptively learned non-verbal cues, the agent completes these tasks much faster than a conventional user-interactive agent and consequently maximizes user satisfaction.

In this chapter, we begin with the motivation behind our approach and then introduce our idea. We also ground our approach with existing approaches that allow machines to learn from body language.

4.1 Towards Natural Forms of Human-Machine Interaction

Human communication is intuitive and sustainable, allowing us to form effective teams that collaboratively work towards solving complex problems. The process of human communication involves both non-verbal cues and speech. In the near future, we need to form similarly effective teams with intelligent

machines, so that we can tackle more challenging problems. Ultimately, we hope to amplify our cognitive and physical capabilities through machines. As a first step towards achieving this grand prize of intelligence amplification, we need to design approaches that allow people to communicate naturally with machines, where the communicative process is as intuitive to that of human communication.

Ideal forms of human-machine interaction needs to enable machines and their users to communicate naturally, similar to that of human communication. In this thesis, we focus on using body language to improve human-machine interaction, similar to how these cues are used in human communication. People naturally produce different non-verbal cues, without much effort, during the process of communication (Frith, 2009; Pezzulo et al., 2013; Scheider et al., 2016). Particularly, these subtle cues differ across people and are usually conditioned upon their current situation or environment. These cues are produced as communicative signals and influence the behavior of their observers. For example, while giving a talk to an audience, we usually make accurate judgments about what the audience is currently experiencing or understanding and use these judgments to adapt our talk. Our ability to perceive others' intentions and motivations or their levels of satisfaction from their body language is fundamental for human communication (Adolphs, 2002).

Similar to human communication, people also produce subtle cues during their interactions with their machines in human-machine interaction domains. More importantly, these cues are produced unintentionally without any cognitive effort. These cues are indicative of their satisfaction towards the machine's behavior (Abdić et al., 2016). As a natural direction for improving human-machine interaction would be to use these subtle cues to teach machines perform a collaborative task.

4.2 Our Approach: Learning from Prospective Body Language Cues

As described previously, in this thesis, we introduce a human-machine interaction approach that allows people to teach their machines through subtle body language cues. Particularly, these non-verbal cues are produced unintentionally by the user interacting with their machine, where the machine is learning to perform a collaborative task according to the user's preferences.

The body language cues produced during ongoing interactions are indicative of the user's satisfaction towards their machine's behavior. The machine learns to associate these cues with their user's satisfaction, which is measured in terms of occasional explicit feedback. Over time, the machine learns to predict its future rewards it would receive from the user directly from their body language. As the machine learns to make predictions about its future rewards from the user's body language, it can adapt its subsequent behavior in a given task even before the user generates an explicit feedback, thereby maximizing user satisfaction. As these cues are used by the learning agent to forecast its future rewards, we call them *prospective body language cues*.

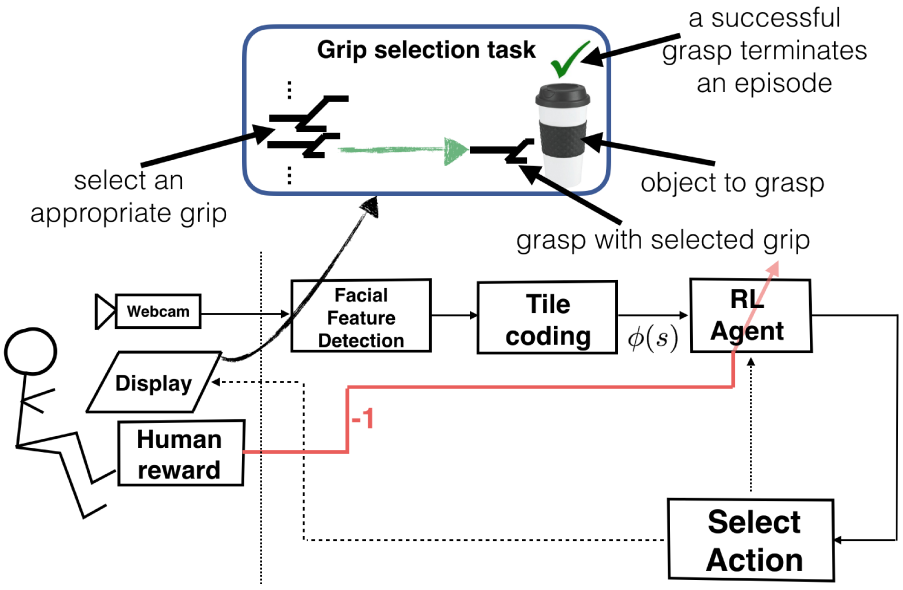


Figure 4.1: A human-machine interaction setup where the agent learns from prospective body language cues to complete a given task. Here, the agent learns to solve a task from facial expressions of the user.

Our human-machine interaction approach of learning from prospective body language assumes that a user is interacting with an implemented learning agent in a human-machine collaborative task. An instance of this setup is shown in Figure 4.1. The goal for both the user and the agent is to successfully complete the given tasks. Specifically, the goal for the learning agent is to figure out the right sequence of actions to take in order to complete the given tasks as fast as possible, and as a result, maximize user satisfaction. The goal for the user is to guide the learning agent in completing these given tasks.

The human-machine interaction setup, which is used in this thesis, involves a user interacting with an agent. The user observes the agent's actions and interactions within an environment as it learns to perform the given collaborative task. The interactions from the user are in the form of non-verbal cues and occasional negative rewards. The agent observes these cues throughout the course of the task and receives these occasional rewards. The process of providing a negative reward to the interacting agent is called an *explicit feedback* and is achieved by the user pressing a button on the keyboard. We assume that, in real-world collaborative tasks, it is easier to provide evaluative feedbacks in the form of body language than generating explicit feedback.

During the interactive task, the user produces non-verbal cues that are indicative of the agent's performance in the task and occasionally provide explicit feedback. As the agent observes these subtle cues and receives these explicit feedback, it learns to associate these cues with the future rewards. Specifically, the agent learns a value function that maps the user's body language to its future rewards. As a result, the agent learns to predict user satisfaction which is measured in terms of the explicit feedback. Over time, the agent learns to use these predictions to adapt its behavior before it receives explicit feedback from the user. In short, the agent perceives the user's body language cues as evaluative signals, critiquing and subsequently influencing the agent's behavior so as to complete the task faster and according to the user's preferences. This directly maximizes user satisfaction because the agent learns to quickly complete the task as preferred by the user.

Learning the meaning of these non-verbal cues and learning an appropriate

behavior based on these cues is occurs together and is handled by conventional temporal-difference learning methods. Our method implicitly involves two processes: first, the agent learns to the meaning of different non-verbal cues in the form of a value function, and second, uses these predictions to learn an appropriate behavior that maximizes its rewards. Overall, our method allows people to tea ch their machines using whatever subtle body language they produce during interaction. This is a general approach and encompasses any subtle non-verbal cue. These cues are associated with their future rewards and learned as a value function. As these cues forecast the future rewards received by the learning agent, these are called prospective body language cues and we say that the agent is *learning from prospective body language*. However, for the experiments in this thesis, we restrict our approach to the subtle cues produced in the form of facial expressions by the user while interacting with their machine and we call this restricted approach *face valuing*, as the agent learns a value function over the user's facial expression.

4.3 Existing Approaches that Use Body Language

In the previous sections, we described our approach that allows people to teach their machines using subtle body language cues, where these cues inform the agent about its future rewards. However, using body language to teach machines is a natural idea in the field of human-machine interaction, and naturally, many existing approaches have attempted at improving human-machine interaction through body language. In this section, we describe the fundamental ideas that exist within the field of human-machine interaction approaches that use body language, in one way or the other, to train machines. In the next section, we contrast these with our approach.

4.3.1 Body Language as Instructions to Machines

The primary goal of a machine in a human-machine interaction task is to learn to perform a task according to the user's preferences. A simple way

of teaching the machine with body language to perform a task is to map different non-verbal cues to different instructions to the machines. In these cases, the user would need to produce the appropriate non-verbal cue so that the machine executes the right instruction. This is the overall idea of the existing approaches described in this section.

Faria et al. (2007) designed a user interface that can be operated using the user's facial expressions. Their approach involves mapping different facial expressions to control signals. The facial expressions are recognized from the user's face that is observed through a simple webcam from a portable laptop, which sends control signals to the wheelchair. In their approach, the user is required to produce a particular facial expression in order to move the wheelchair in a specific direction. The mappings between the facial expression cues and instructions are predefined by the system designer, implying that the users are required to remember these mappings and translate their expressions accordingly to effectively control these wheelchairs.

Similar approaches were also developed by many researchers, the popular ones are listed here: Wei et al. (2009), Lievesley et al. (2011), Faria et al. (2012), Rechy-Ramirez et al. (2012), Tanaka et al. (2005), Galan et al. (2008), Ferreira et al. (2007), She et al. (2014), Cruz et al. (2015). Many of these approaches used different methods for extracting facial expressions and/or EEG signals from the user. However, all of them still rely on the system designer to map these cues to different control signals. More importantly, these approaches require the user to remember the body language mappings and translate their cues to effectively control the system.

Reis et al. (2009) designed a comprehensive framework that allowed users to control their wheelchairs through a combination of modalities, like voice, facial expressions, head movements, keyboard, joystick etc. More importantly, their framework allowed the users to program their own sequence of cues to a control signal, prior to the operation of the wheelchair, as opposed to approaches that relies on the system designer to map cues to commands.

4.3.2 Body Language as Rewards to Machines

In this body of research, the general idea is to map non-verbal cues to reward signals. The idea here is to shape the existing reward channel of the learning agent using user's body language. More specifically, a specific body language cue results in producing a certain reward to the agent. Many of the approaches in this line of research rely on supervised learning approached to classify different non-verbal cues and the mapping of these cues to different reward signals is achieved by the system designer. Some approaches use verbal instructions as rewards instead of non-verbal cues.

There are no research works, to the best of my knowledge, that uses body language as reward signals to machines. However, this is an obvious idea, that is fundamentally same as interacting with machines through user-generated rewards. As it is similar to that of learning from user-generated rewards, this research direction would also have similar issues, some of which are described below.

The first issue that arises when teaching a machine using user-generated rewards is that the reward distribution produced by the user changes as the experiment progresses. Specifically, during the initial part of the experiment, the user provides more rewards and actively continues this until the agent learns a desired behavior. After the agent learns to behave in an appropriate manner, the user does not feel the necessity to continue with the interaction process. Ideally, the user should not be providing rewards when the agent behaves in the right way. However, reinforcement learning algorithms cannot handle this, driving the agent to explore different actions and as a result deteriorates its behavior. In order to generate a stable stream of user-generated rewards, Knox and Stone (2009), introduced the TAMER framework, where a supervised learning model of the user-generated rewards is first learned from the user and then used in place of the user. This stabilizes the interacting agent, executing a behavior that is appropriate with the user. However, when the task changes or when the agent is faced with a new task, then the user needs to intervene and reprogram the supervised model in order to handle this

change of task.

An important requirement, and another potential limitation, in this line of research is that complex training datasets need to be created in order to recognize subtle cues, that are specific to the user.

4.4 Comparison to Existing Human-Machine Interaction Approaches

In the field of human-machine interaction, the existing ideas either view body language as instructions or as reward signals to the machines. More importantly, the approaches resulting from these ideas assume that people produce similar non-verbal cues to convey their reactions towards the machines. This is a flawed assumption to make, because different people produce different body language to convey what they feel about their current situation. Designing approaches for human-machine interaction based on this assumption would require the users to translate their body language to conform with their machine's design. Using such approaches in real-world human-machine interaction tasks, like the ones involving prosthetic limbs or autonomous cars, would require the user to focus some part of their attention to effectively interact with the machine. Consequently, this cognitively imposes on the user, making them to lose focus on what is important at hand. Ideally, this scenario needs to be turned around. The machine needs to be adapting to the user, learning from whatever subtle cues they produce during the process of interaction, rather than requiring the user to adapt to the machine.

As opposed to these existing approaches, the approach we introduced here does not assume that all people produce similar body language. More importantly, our approach learns the meaning of the different non-verbal cues directly from ongoing interactions with the user. The machine learns to associate whatever cues with user-generated rewards, and over time, learns to predict the user feedback from their body language. The meaning of these non-verbal cues are grounded in terms of the later rewards the machine receives from the user. As the machine learns to predict its future rewards

from the user, it can then adapt its behavior in order to maximize its rewards and thereby maximize user satisfaction. This is different from any existing approaches to human-machine interaction because our approach learns the meaning of non-verbal cues from the user.

Chapter 5

Using Sarsa To Learn From Prospective Body Language

In the previous chapter, we introduced our idea of teaching machines through body language cues. Particularly, the machine learns to pick up whatever subtle cues people produce during the process of interaction and associate these cues with their user satisfaction. By learning a value function that maps these non-verbal cues to future rewards, the users can teach the machines to act according to their preferences, thereby teaching the machine to complete the given tasks quickly. In contrast to our introduced approach, the existing approaches that learn from body language usually rely on the system designer to map these different cues to instructions that are executed by the machine. Some methods map these cues as reward signals to the learning machine. As these mappings to rewards or instructions are non-adaptive, they require the user to translate their body language to effectively use such machines, and as a result cause significant cognitive load on the interacting user.

In this chapter, we implement our approach using a standard action-value learning algorithm, namely Sarsa, and compare its performance with another agent that learns only from user-generated rewards. Specifically, we implement our approach to learn from subtle facial expression cues produced by the user while interacting with the machine whereas the conventional agent relies only on the user-generated rewards to figure out the right approach to complete the given tasks.

Though our approach is general enough to learn from any body language

cue, here, we restrict ourselves to facial expression cues of the user and this approach is called face valuing.

5.1 Implementation using Sarsa

The face-valuing approach involves a user interacting with an implemented agent which is learning to complete a given human-machine collaborative task. In order to complete this task quickly, the user needs to communicate their preferences to the machine. Particularly, the agent needs to figure out the right action to take, according to the user in order to complete the task.

A simple way of realizing this approach as an interactive agent is by using a standard reinforcement learning algorithm called Sarsa. This is a simple and popular action-value learning algorithm. Sarsa extends temporal-difference learning to learn an action-selection policy which can be used to pick actions that maximize rewards achieved by the learning agent. For a comprehensive description about this algorithm refer to the textbook written by Sutton and Barto (2017).

Experiment Setup. The face-valuing approach involves a setup where a user is interacting with a virtual agent, who observes the agent through a display. At every time step, the agent receives a new raw image from a standard webcam. This image is assumed to contain the user's face and is processed into features through a standard facial detection algorithm, which is then tile-coded to form the feature representation for the agent. These features approximate the current state of the agent and is generated at each time step from the facial expression cues of the user. The reinforcement learning agent learns to associate and predict its expected rewards from these feature representations, learned with occasionally generated explicit corrective feedback from the user. Based on these action-value predictions, the agent learns to take actions that maximize user satisfaction (i.e., minimize the explicit feedback received over time). Through this approach, the agent learns to take actions prior to the user generating any explicit feedback.

In subsequent sections, we describe the process involved in constructing

the state representation for the face-valuing agent and the algorithm as pseudocode. In later sections, we describe the simulated human-computer collaborative task that is used in our experiments and report our results obtained from the face-valuing Sarsa agent.

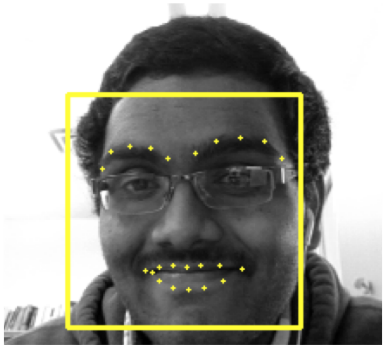


Figure 5.1: State representation for face-valuing agent: Features extracted from a user's face.

5.1.1 State Representation: Processing Raw Images into Facial Expression Features

The state representation is perhaps the most important component to a reinforcement learning agent. It allows the agent to identify its current state within an environment. By reliably identifying its current state, the agent learns to take actions so as to maximize its reward signal from the environment. More generally, we can describe the state representation as the "eyes" of the agent, which enables it to identify and represent different states that it observes during its interactions with the environment.

The state representation for the agent needs to be designed so that it can identify various states reliably from its features. This state space of the face-valuing agent allows it to learn its expected reward or punishment grounded in the facial cues of its user from ongoing interactions. For our experiments, we use a simple facial feature detection algorithm which extracts interesting feature points from the user's face. The features extracted from a user's face is shown in Figure 5.1.

The face-valuing agent observes a frame containing the user's face obtained from a webcam. This image frame, which is assumed to contain the user's face,

is processed and 68 facial landmark points are extracted using a popular facial landmark detection algorithm (Kazemi et al., 2014). These facial landmark points are simple two dimensional coordinates over the raw input frame that denote the position of certain special locations of the user's face, as shown in Figure 5.1. Particularly, these landmark points localize salient regions of the face, namely the eyes, eyebrows, nose, mouth and jawline. These points are also called key points and are usually used for aligning multiple images taken from different viewpoints. However, these are sufficient for our purpose and our experiments as they provide a crude approximation of the facial expression produced by a user.

At every time step, the agent receives a raw frame from the webcam, from which it extracts these 68 keypoints. These keypoints are normalized by subtracting out their mean value and dividing them by their standard deviations. Out of these 68 key points, 23 points are selected, which excludes the features corresponding to the jawline. Each of these 23 points, are individually tile-coded with 4 tilings. Each of these tilings are of size 10×10. This results in a feature vector of size 9200 and this vector forms a part of the state space. It is referred as $\phi(s)$.

Facial landmark points are chosen as features for representing facial expression cues because these points are easy to compute from raw images in real-time, without involving much computation. This is appealing because we are concerned with building a real-time human-computer interaction approach that allows users to communicate with their learning agents through ongoing interactions. The form of communication involves using different facial cues, with its meaning learned from ongoing interactions involving infrequent explicit feedback.

5.1.2 Sarsa Learning Algorithm

Reinforcement learning algorithm is the core component of our face-valuing approach. The facial feature detection algorithm is used to construct the state representation from the user's facial expression cues, but the important learning process happens through the following reinforcement learning algorithm.

Particularly, the learning process is responsible for associating facial expression cues to the future rewards experienced by the agent while operating within the simulated environment.

In this face-valuing approach, user satisfaction is formalized as an action-value function of a learning agent and is learned through Temporal-Difference learning.

Algorithm 4 Face Valuing: Implementation using Sarsa

INPUT: $\alpha, \lambda, \gamma, \boldsymbol{w}_{init}$
  $\boldsymbol{w} \leftarrow \boldsymbol{w}_{init}$ ▷ $\boldsymbol{w}$ is the weight vector for the action-value function
  $\boldsymbol{e}_w \leftarrow 0$ ▷ $\boldsymbol{e}_{\boldsymbol{w}}$ is the eligibility trace for the action-value function
  for num. of episodes **do**
    obtain initial state S
    construct the features $\phi(S)$ corresponding to initial state S
    (from the user's face, as described in Section 5.1.1)
    select action A based on state features $\phi(S)$ (for example, ϵ-greedy)
    $\psi \leftarrow$ features corresponding to S, A
    while S is not terminal **do**
      take action A, observe next state S' and reward R
      construct the features $\phi(S')$ corresponding to the next state S'
      select action A' based on $\phi(S')$
      $\psi' \leftarrow$ features corresponding to S', A'
      $\delta \leftarrow R + \gamma \boldsymbol{w}^\top \psi' - \boldsymbol{w}^\top \psi$
      $\boldsymbol{e}_w \leftarrow \gamma \lambda \boldsymbol{e}_w + \psi$
      $\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha \delta \boldsymbol{e}_w$
      $S \leftarrow S'$
      $A \leftarrow A'$
      $\psi \leftarrow \psi'$
    end while
  end for

At every time step, the images obtained from the webcam are processed into features corresponding to the facial expression cues of the user, who is interacting with the agent. Using these features, the agent estimates its likelihood of picking different actions. The learning agent consists of a set of parameters $\boldsymbol{w} \in \mathcal{R}^n$, which is incrementally updated from rewards received by the agent. This set of parameters, when multiplied with the features corresponding to a state-action pair, gives an estimate of the expected reward that would be received by the agent by picking this corresponding action. As the algorithm

learns the utility of each state-action pair, this is called an action-value method and one such action-value algorithm, called Sarsa, is used for constructing the face-valuing agent. This is described as pseudocode in Algorithm 4.

5.2 Grip-Selection Task

We introduce a grip-selection task that is inspired from a difficult problem from the prostheses domain and use this task to evaluate our face-valuing approach. In this task, the objective for the agent is to select an appropriate grip pattern for a given object and use this grip to successfully grasp the object.

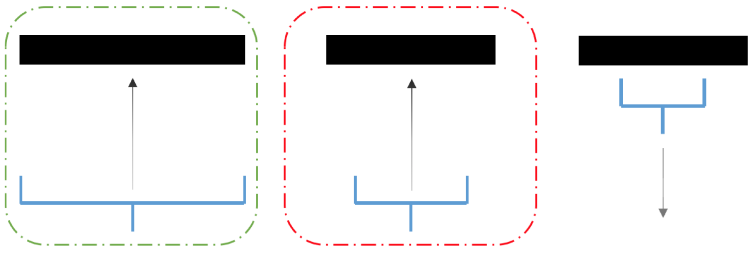


Figure 5.2: Grip-selection task

At the beginning of the experiment, an object is randomly created and shown to the user. This object is located at a distance from the location where the agent's grip is spawned. This initial location is called the "grip-changing station" and here the agent can select from a set of grips. The objective for the agent is to complete this task, by selecting an appropriate grip at this grip-changing station, then move towards the object's location and complete a grasp of this object. A successful grasp of the given object using an appropriate grip completes an episode. The appropriate grip for completing a grasp of this object depends on the user's preference — there could be many different grips available to the agent and the right one here depends on the user. This is similar to many real-world human-machine interaction tasks, like prostheses and self-driving cars, where the goal for the agent is to behave according to the user's preferences.

The task consists of a set of n grips and m objects. Depending on the experiment setting, there could be many possible grip-object combinations that

could complete an episode successfully. However, the correct grip-object combination depends on the user who ultimately evaluates the agent's behavior.

The task is formulated as an undiscounted episodic MDP with a 0 reward at every time step from the environment. Also, there is no reward from the environment for a termination of an episode. The only available reward signal is from the interacting user: by pushing a button, the user can produce a reward of -1 to the agent for the corresponding time step.

5.2.1 State Space

The state space for the agent without face valuing consists of the given object's ID and the agent's current selected grip ID, along with a bias term. Specifically, the ID of the object that is displayed in an episode is one-hot encoded to form a binary valued vector of size m, where m is the total number of objects available in an experiment setting. Similarly, the ID of the agent's current selected grip is one-hot encoded to form a binary valued vector of size n, where n is the number of grips available to the agent at its grip-changing station. Both these vectors are concatenated along with a bias term with a value of 1 to form the state's feature vector of size $m + n + 1$. This forms the state space for the agent without face valuing.

For the face-valuing agent, the state space consists of the features corresponding to the user's facial expression cues. The facial landmark points from the user's face are extracted, at each time step, and processed into a vector of size 9201. This vector is the feature representation corresponding to the user's facial expression cues and represents the state space for the face-valuing agent.

5.2.2 Action Space

The complete action space for the agents (with and without access to face valuing) consists of the following actions: $\{grip_1, grip_2, \cdots, grip_n, \uparrow, \downarrow\}$, where the first n actions results in selecting the corresponding grip, from a set of n grips available within an experiment setting. The remaining two actions move the agent towards and away from the given object.

Though there are $n+2$ actions, all of them are not available to the agent at all times. The actions available to the agent depends on its position relative to the object and grip-changing station. When the agent is in the grip-changing station, the available actions are $\{grip_1, grip_2, \cdots, grip_n, \uparrow\}$ whereas when the agent leaves the grip-changing station, $\{\uparrow, \downarrow\}$ actions are available. In other words, the agent can change its grip only at this grip-changing station. Once it leaves this station, the agent cannot change its grip unless it returns back to this position.

5.2.3 User-Generated Rewards

As described previously, there are no reward signals from the environment. The sole source of reward is from the interacting user. Specifically, when the user pushes the reward button, the agent receives a -1 reward for the corresponding time step. Moreover, on pressing this reward button, the agent loses all its actions except $\{\downarrow\}$ until it reaches the grip-changing station.

The agent observes the state space *once every three-tenth of a second* and takes an action at every time step. The agent, however, has the freedom of choosing the same action for many consecutive time steps which allows the user to observe the agent's action and then expressively respond to the agent.

5.3 Experiments and Results

The experiments and results in this section involved a comparison between two types of learning agents: one with access to the task-specific features and the other with access to face-valuing features. The agent which uses only the task-specific features does not have access to face-valuing features and is a simple user-interactive approach that learns from human interactions in the form of rewards.

The first set of experiments consisted of varying the number of object and grips available within a task and comparing the learning performances of the agents. In the second set of experiments, both these agents had limited number of grips available within the task, but a new object was generated for

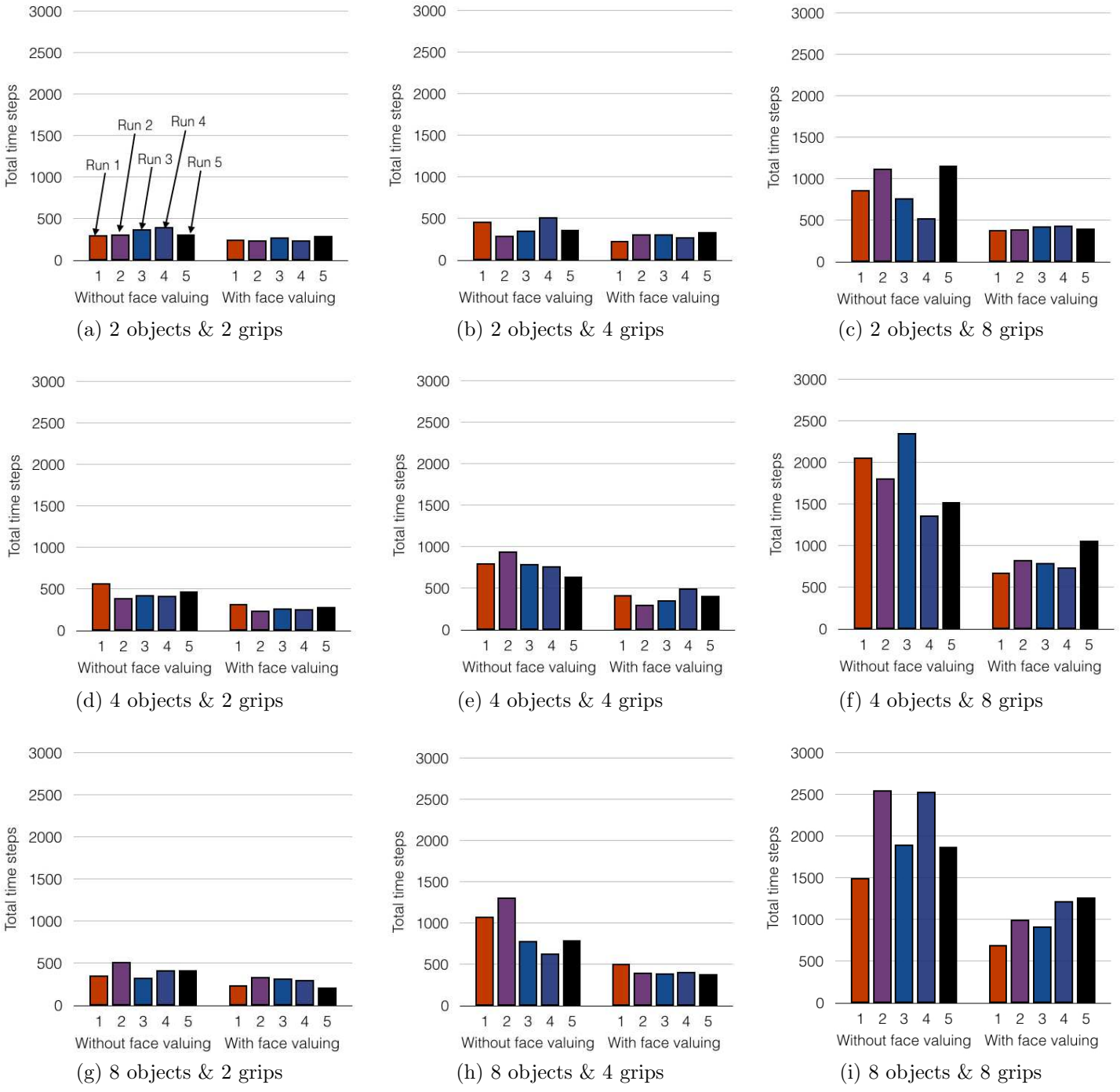
each episode. More importantly, no object was repeated during this second experiment, essentially making each episode a new task to the learning agent. This is a realistic setting of a real-world grasping task, where there are infinitely many objects available in the world and they need to be picked up using a limited set of grips

Both the agents used in these two experiments were constructed using the Sarsa learning algorithm. Specifically, two Sarsa agents, one with face-valuing features and the other with task-specific features, were evaluated. Throughout the course of the experiment, the user does not know the type of agent that is currently interacting. In other words, the experiments were conducted in a blind manner and consisted of 15 episodes.

5.3.1 Experiment 1: Multiple Grip and Object Setting

In this first experiment, we varied the number of available grips and objects available within the simulated task and compared the learning performances of different user-interaction agents. Specifically, for different object and grip settings, we compared the learning performances of an agent with access to task-specific features and an agent with access to face-valuing features.

Each experiment consisted of 15 episodes. At the beginning of every episode, an object and a grip, randomly selected from the set of available objects and grips, is displayed to the user through the display. During an episode, the object remains fixed whereas the agent can change its grip from one of its available grips. The episode terminates only on a successful grasp using an appropriate grip.



(a) 2 objects & 2 grips

(b) 2 objects & 4 grips

(c) 2 objects & 8 grips

(d) 4 objects & 2 grips

(e) 4 objects & 4 grips

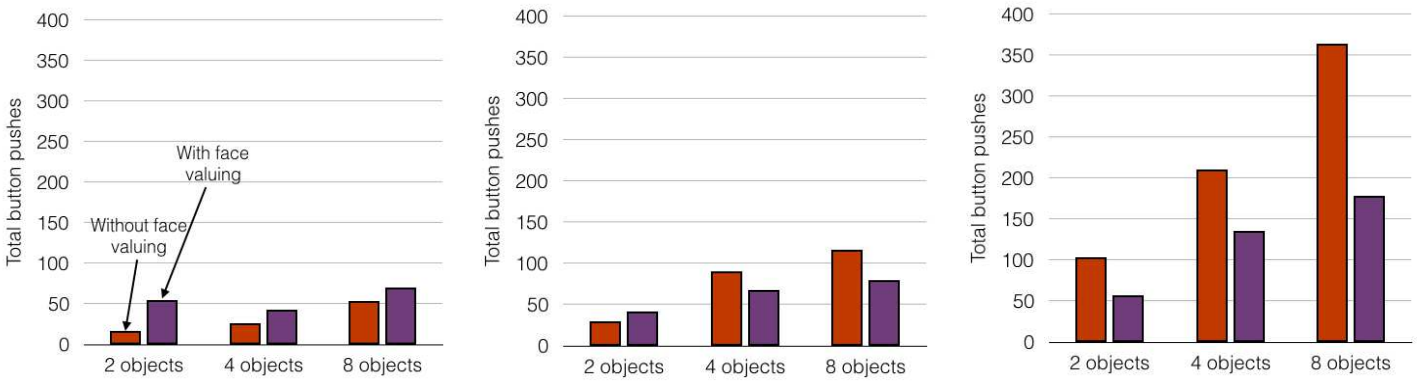
(f) 4 objects & 8 grips

(g) 8 objects & 2 grips

(h) 8 objects & 4 grips

(i) 8 objects & 8 grips

Figure 5.3: Total time steps taken for different grip and object settings by Sarsa learning agents.



(a) 2 grips setting (b) 4 grips setting (c) 8 grips setting

Figure 5.4: Total number of user-generated rewards (i.e., explicit feedback) provided to the learning agents.

The plots of total time steps taken and total user-generated rewards accumulated by the agents during this experiment are shown in Figures 5.3 and 5.4. The plots in Figure 5.3 represents the cumulative time taken by a learning agent to complete a successful grasp over multiple episodes. The plots in Figure 5.4 displays the number of times the user provided an explicit feedback to the learning agents. Both these graphs were generated from the same user experiments, conducted in a blind manner. A perfect agent that knows its user's intentions would receive no user-generated explicit feedback in all these settings. Moreover, it would take exactly 11 time steps for completing an episode: one time step for picking the right grip and ten time steps for moving towards and grasping the given object.

From the plots in Figure 5.3, the agent with access to face-valuing features quickly adapted with respect to the user's preferences in all the experiment settings. The face-valuing agent achieved this performance with significantly less number of user-generated feedback than that of the agent with access to task-specific features.

5.3.2 Experiment 2: Infinite Object Setting

This second experiment was designed to show the performance improvement obtained from face valuing in a difficult and a realistic task. In this experiment

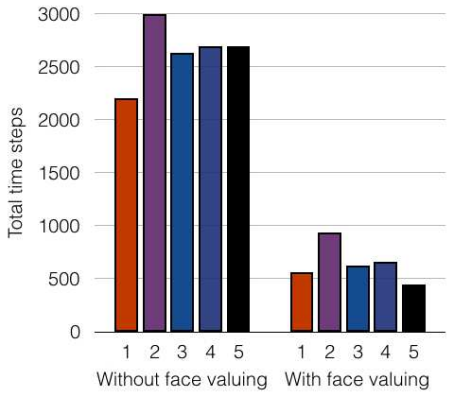


Figure 5.5: Total time steps taken to complete the infinite objects setting by Sarsa learning agents.

setup, a new object is generated at the beginning of every episode in the grip-selection task. Specifically, each episode uses a new object, making each episode a new task to the learning agent. The objective for the learning agent here is to complete the grasp of this given object by selecting a grip from its limited set of grips. This setup allows in exploring the ability of a face-valuing agent to leverage from previous learning experiences to address new or changed tasks.

It is important to note here that, in this experiment setting, an agent with task-specific features cannot learn the user's preferences from task-specific features. This is because each episode is a new task to the learning agent. The task-specific agent can solve this task only through trial and error: by trying out each grip for each object.

The graphs in Figure 5.5 show the total time steps taken by the learning agents. From this figure, it is quite clear that the face-valuing agent is much quicker in adapting to its user's preferences, which are communicated as facial expression cues. Particularly, the body language of the user does not change much throughout the task and this allows the face-valuing agent to reuse its previous learning to complete unseen tasks with new objects. This is in contrast to the task-specific agent, which cannot reuse its previous learning because of its state space. More importantly, these results confirm the conclusions drawn from the previous experiment, which is that face-valuing

agents can complete human-machine collaboration tasks much faster than the agent using task-specific features because it relies directly on the user's body language for figuring out the correct sequence of action to take.

A standard t-test between the learning performances of the agents with and without face valuing gives a value of 0.0002, meaning that the results are statistically significant. These values were obtained from a paired t-test using a two-tailed distribution.

5.4 Discussion

Several studies have shown that users, to a certain extent, are willing to amplify their cognitive and physical abilities through machines. For example, in medical domains, it is common for people with amputations to extend their capabilities or overcome their limitations by forming partnerships with machines (Williams, 2011). However, the current technologies are unsustainable and unscalable to form successful human-machine partnerships. The fundamental issue that hinders in achieving this goal is that the existing technologies are not capable of identifying and adapting to the changing personal preferences of their users. Instead, these technologies expect the user to translate their communicative signals according to the system design, which leads to user frustration towards the system. This issue serves as a serious bottleneck to intelligence amplification. The human-machine interaction approach, introduced in this thesis, serves as the first steps towards achieving this ambitious goal of intelligence amplification.

Though our experiments were simulated, we believe that our approach can be much more valuable in a realistic robot setting — we expect a robot's behavior to elicit more expressive facial feedback from the user, in comparison to our simple simulated domains. More importantly, these expressive feedback would serve as powerful features to a face-valuing agent.

The grip-selection task is a task where the learning agent needs to figure out the goal from its ongoing interactions with a user. The agent, operating within this task, can be termed as a *goal-seeking agent* (Pilarksi et al., 2015).

In order to demonstrate the significance of our approach, we conducted two sets of experiments: the first one involved multiple object-grip settings on the grip-selection task. The second experiment involved in generating a new object for each episode. This infinite object setting is pertinent to real-world scenarios, where there are uncountable number of objects that can be grasped from a limited set of grip patterns.

In our experiments, we compare learning agents that operate with different state spaces. Particularly, one agent operates with a state space that identifies the given object and the current selected grip by the agent. Using this task-specific state representation, the agent can learn to identify the correct object-grip combinations from user-generated rewards. The agent learns to pick the correct grip *after* trying out each possible grip for a given object and this policy is learned through user-generated rewards. For a face-valuing agent, the state space consists of the user's facial expression cues that is processed at each time step. This state space conveys the user's preferences or intentions through facial expressions. The agent using this state space learns to associate the meaning of these expressions with its experienced rewards. More importantly, this state space lets the agent figure out the user's intents irrespective of the task as this is independent of the task. This allows the agent to learn faster in a task without trying out every possible grip. Specifically, this significantly reduces the amount of experience required by the agent to learn a behavior according to the user's preferences.

The results from the first user experiment (Section 5.3.1) suggests that the face-valuing agent learns to adapt quicker to its user's preferences in a given task, when compared to an agent without access to face valuing. Moreover, from the plots comparing distribution of user-generated rewards, it can be observed that the face-valuing agent achieves this performance with significantly fewer number of explicit feedback from its user.

From the second user experiment (Section 5.3.2), we empirically show a scenario where a conventional agent can fail. From both user experiments, we can see that the face-valuing agent successfully adapts and completes one episode after another by relying on helpful clues in the form of facial expres-

sions, specifically by learning a value function over these facial expressions. On the other hand, the agent without face valuing can rely only on the user-generated reward signals for identifying the correct grip for a given object. In tasks with many possible outcomes, for example a task with many object-grip combinations, relying only on the reward channel to communicate user intents is not a scalable or sustainable approach.

From our experiments, we can clearly see that the face-valuing approach allows the agent to perceive an evaluative feedback directly from the user's facial expression cues, thereby allowing the agent to figure out the user's preferences in a given task. Our results suggests that, by learning a value function over the user's facial expression cues, the agent can adapt quickly to the user's preferences, requiring less explicit corrective feedback from the user. The learning process took place as follows: during the initial phase of the experiment, the agent used the occasionally provided explicit feedback to learn value function over the user's facial expressions. These cues served as useful clues about the future rewards that the agent would receive by following its current behavior, thereby guiding and adapting its behavior.

Limitations of using action-value methods in our approach. The approach we introduced in this thesis involves learning from whatever subtle cues the user produces while interacting with an agent. It is important to note here that these body language cues and their association with future rewards are task independent, meaning that this learned function can be used in different tasks than the ones used for initial training. Particularly, we can think of scenarios where the user interacts with a simulator where the agent learns to pick on different cues specific to the user and later share this learning with a real-world machine, like a prosthetic arm or an autonomous car. Sharing and transferring the learning from a user across tasks improves the overall experience across these tasks, as it would involve much less training when compared to learning from scratch.

An action-value implementation of face valuing does not exactly conform to this idea of transferability across different domains and tasks, because of the fact that the values and policy are closely coupled as action-values. It would

be desirable to design an approach which learns a value function based on the user's body language independent of the action-selection policy, so that we can share this value function across domains and learn the policy from scratch.

Chapter 6

Using Actor-Critic To Learn From Prospective Body Language

In the previous chapters, we introduced our idea for teaching machines through prospective body language and an action-value approach for achieving this idea as an implemented system. Also, we introduced the process of constructing the state space for a face-valuing agent, along with a simulated grip-selection task for evaluating our approach. Using this simulated task, we conducted experiments comparing the face-valuing agent with a conventional user-interactive agent.

In this chapter, we introduce an actor-critic architecture for implementing our idea for learning from body language. Specifically, using a Sarsa agent estimates the utility of each action that is available in a given state. This means that the method relearns information when they can actually be reused across these actions. More importantly, an action-value approach does not easily allow in transferring the learned value function across domains. To address this specific issue, we introduce an actor-critic approach that reduces the number of parameters that need to be learned to solve a task and easily allows in sharing this learned value function across multiple domains, without losing its advantage over a conventional user-interactive agent that learns from user-generated rewards.

6.1 Implementation using an Actor-Critic Method

The setup used in the following experiments are the same as described in the previous chapter. The approach involves a user interacting with an implemented learning agent in a human-computer collaborative task, where the user needs to communicate their preferences effectively in order quickly complete the given tasks. Here, user preferences are over the grips for the agent in order to complete the task quickly.

In the following section, we describe the actor-critic architecture that we use for implementing our face-valuing agent.

Actor-critic methods are popular reinforcement learning approaches that consists of two separable learning modules, namely the actor and critic. The actor is responsible for learning a policy that is used for selecting an action for a given state and the critic critiques the actions selected by the actor module. Specifically, the critic generated the error signal that drives learning in both these modules.

The actor learns a function that produces the likelihood of choosing an action for a given state within the environment. The actor usually increases the likelihood of picking an action if it produced a higher than average TD-error for a given state and otherwise it reduces its likelihood. The critic module learns the state-value estimates for the states observed by the agent during its interaction with the environment. A standard TD(λ) algorithm is used for learning these state estimates.

In this face-valuing approach, user satisfaction is formalized as the critic's value function of a learning agent and is learned through temporal-difference learning methods. The actor module learns a policy that chooses actions according to the critic's estimate for a given state.

In the actor-critic method, the action selection behavior is learned directly from rewards received over time, instead of learning action-value estimates and then computing an action-selection policy. Specifically, the actor module learns an action-selection policy π which is parameterized by $\boldsymbol{\theta} \in \mathcal{R}^n$. This policy $\pi(s|\boldsymbol{\theta})$ is like a function that takes in the state as input and produces the

Algorithm 5 Face Valuing: Implementation using Actor-Critic Method

INPUT: $\alpha_v, \alpha_\pi, \lambda, \gamma, \boldsymbol{w}_{init}, \boldsymbol{\theta}_{init}$

$\boldsymbol{w} \leftarrow \boldsymbol{w}_{init}$ ▷ $\boldsymbol{w}$ is the weight vector for the critic

$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}_{init}$ ▷ $\boldsymbol{\theta}$ is the weight vector for the actor

$\boldsymbol{e}_w \leftarrow 0$ ▷ $\boldsymbol{e_w}$ is the eligibility trace for the critic

$\boldsymbol{e}_\theta \leftarrow 0$ ▷ $\boldsymbol{e}_\theta$ is the eligibility trace for the actor

for num. of episodes **do**

  obtain initial state S and $\phi(S)$ from facial features

  while S is not terminal **do**

    select action A for state S (using $\psi(S)$)

    take action A, observe S' and R

    $\delta \leftarrow R + \gamma \boldsymbol{w}^\top \phi(S') - \boldsymbol{w}^\top \phi(S)$

    $\boldsymbol{e}_w \leftarrow \gamma \lambda \boldsymbol{e}_w + \phi(S)$

    $\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha_v \delta \boldsymbol{e}_w$

    $\boldsymbol{e}_\theta \leftarrow \gamma \lambda \boldsymbol{e}_\theta + \nabla_{\boldsymbol{\theta}} log[\pi(A|S, \boldsymbol{\theta})]$

    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha_\pi \delta \boldsymbol{e}_\theta$

    $S \leftarrow S'$

    $\phi(S) \leftarrow \phi(S')$

  end while

end for

probability of each action available within the environment. The actor-critic method also consists of a critic module that critiques the actions selected by the actor. This critic is parameterized by a different set of parameters $\boldsymbol{w} \in \mathcal{R}^n$. Usually, a standard TD learning algorithm (as described in the background section), is used for learning the critic's parameters. Also, this TD-error generated by the critic drives the learning in the actor module. In simple terms, the actor module learns to increase the likelihood of picking an action for a given state when it produces a positive TD-error. Otherwise, this likelihood decreases.

The state representation is usually shared between actor and critic modules. However, here, we choose to use the critic's predictions of the current state as input to the actor module (denoted as $\psi(S)$ in Algorithm 5). Intuitively, it makes sense to pick one action when the state has a larger estimate of expected reward and pick the other action when the state has a smaller value estimate, especially in domains where the action space can be parameterized as a continuous spectrum between two opposing actions. Correspondingly, for

the states whose estimates lie between these two extremes, the actions lying across this spectrum is picked. This is particularly achieved by our modified actor-critic approach and is described as pseudocode in Algorithm 5. However, for our experiments, we use a discrete action space. Furthermore, the potential trade-offs of using these two algorithms for implementing face valuing approach are discussed later.

6.2 Experiments and Results

The experiments and results in this section involved a comparison between two types of learning agents: one with access to the task-specific features and the other with access to face-valuing features. The agent which uses only the task-specific features does not have access to face-valuing features. Throughout the course of the experiment, the user did not know the type of agent that is currently interacting. In other words, the experiments were conducted in a blind manner. Each experiment consisted of 15 episodes.

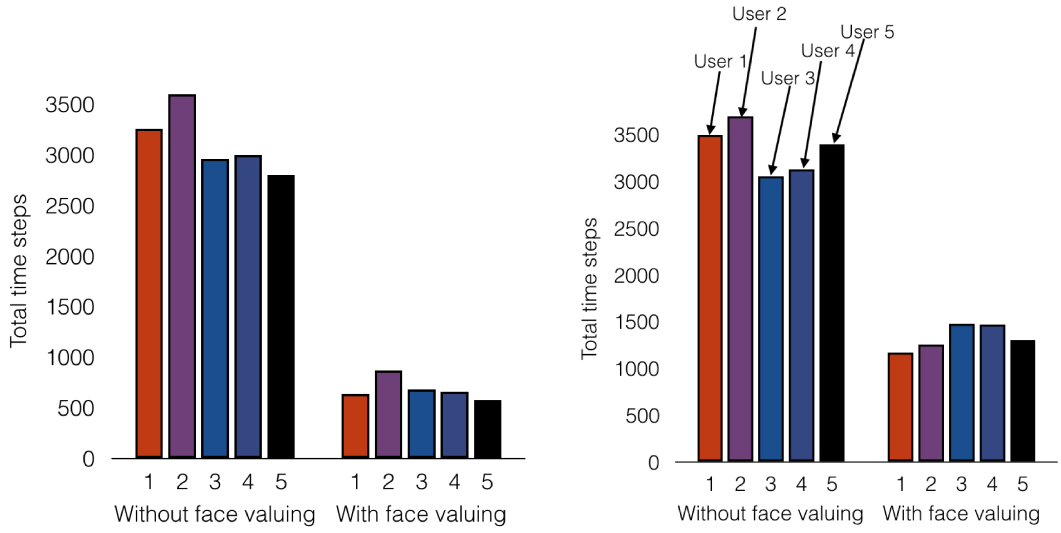


Figure 6.1: *Left* Total time steps taken to complete the infinite object setting by the actor-critic agents. *Right* Total time steps taken to complete the infinite object setting. Experiment performed by 5 different users with no prior experience in this task.

In this experiment, the simulated grip-selection task consisted of a finite

number of grips. At the beginning of each episode, a new object was generated and presented to the interacting user. This object was located at a distance from the location where the agent and its grip are spawned. The objective for the learning agent here is to complete a grasp of this given object successfully by using one of the appropriate grips chosen from this limited set of grips. It is important to note here that the agent can switch its grip only at the grip-changing station. The agent cannot switch its grips when it is away from this grip-changing station.

This experiment was designed to show the performance improvement achieved through our face-valuing approach in a difficult and a realistic task, where the agent needs to grasp a new object that is generated in each new episode. The experiment tests the utility of reusing the agent's previous learning experiences within the grip-selection task.

Each experiment involved 15 episodes and the results obtained from our experiments are reported in Figure 6.1. The figure on the left was obtained from a user who had previous experience interacting with the face-valuing agent and the simulator and the figure on the right was obtained from 5 different users who had no prior experience with the approach or the simulator. The graphs report the cumulative time taken by the corresponding learning agents in completing a set of 15 episodes, each of which involved the agent in selecting an appropriate grip and successfully grasping a new object.

From the plots, it is clearly observed that the face-valuing agent achieves a much better performance that a conventional agent that learns from user-generated rewards. Particularly, the face-valuing agent has the ability to reuse its previous learning experiences to solve new tasks, where this ability arises from the state space constructed using the user's body language cues. On the contrary, the task-specific agent operates with a state space that has no potential to reuse or leverage from previous experiences as each episode generates a new object.

A standard t-test between the learning performances of the agents with and without face valuing gives a value of 0.005 (for Figure 6.1(*Left*)) and a value of 0.02 (for Figure 6.1(*Right*)), meaning that the results are statistically

significant. These values were obtained from a paired t-test using a two-tailed distribution.

6.3 Discussion

In this chapter, we introduced an actor-critic approach for face valuing, which learned a value function grounded in the user's facial expressions, independent of the action-selection policy. The motivation behind this is that, ideally, we would want to share this learned value function across multiple domains as this is task-independent. Particularly, the user's body language would remain the same, have the same meaning, irrespective of the task the machine is performing. We can leverage this insight by sharing the learned value function to different tasks, and as a result, improve the overall user experience with the machine. It is important to note here that this actor-critic approach allows in transferring the learned value function across multiple domains without much effort, and this is not possible in the action-value version of face valuing.

In the actor-critic implementation of face valuing, the policy operates over a state space that is based on the critic's estimate. Specifically, the critic's value estimate is tile-coded using a one dimensional tilecoder and this forms the state representation for the actor module. This is a simple representation of the actor module because, essentially, the module learns to pick one action when the estimate is below a certain threshold. Otherwise, it learns to pick other actions. Such a policy representation is suitable only for certain tasks, like navigation or tasks with binary choices. The representation capability of this actor module is sufficient to learn a policy over a linear spectrum of an action. For example, if the agent is learning to navigate, we can formulate a continuous action space for such a domain where an action at one end of the spectrum can move the agent forward with maximum speed and an action at the opposite end of this spectrum moves the agent in reverse with maximum speed. Similarly, the actions between this spectrum produces similar consequences with slower speeds.

Chapter 7

Discussions and Extensions: From Facial Expressions to Body Language

In this thesis, we introduced approaches that allows users to communicate with their machines through simple and intuitive channels, specifically, through their body language cues. These cues are used to forecast the machine's future rewards and because of this, these cues are called prospective body language cues. We strongly believe that this work would serve as the first steps towards building more natural and high bandwidth channels of communication between humans and machines.

There are many scopes for improving our approach in order to handle different and complex forms of cues generated by the user, thereby moving closer towards achieving intelligence amplification. Making progress in these avenues should produce many successful human-computer collaborations, especially those that are suitable for many real-world domains.

In this chapter, we discuss a few important questions related to our approach and the potential future directions for extending our approach.

7.1 What is Missing in Existing Human-Machine Interaction Approaches?

Human-machine interaction approaches are supposed to enable people in forming effective collaborations with their machines, resulting in people amplifying

their cognitive and physical capabilities. In order to accomplish this ambitious objective, the process of communication between people and machines need to closely follow the principles underlying human communication, so that the process of communication between these domains are comparable.

Human communication is natural and intuitive because it relies on both speech and body language. In this thesis, we focus on using subtle body language cues to improve human-machine interaction, similar to how body language is used in human communication. Particularly, in human communication, body language cues influence the behavior of their observers and are produced involuntarily without much effort. As a first step towards achieving sustainable human-machine interaction, we introduced our approach that allows people to naturally teach their machines using subtle body language cues, that are produced during the process of interaction. As studied in human communication, the body language of the interacting user influences the machine's behavior.

The existing approaches in the field of human-machine interaction are not comparable to the process of human communication. Specifically, these approaches do not allow people to communicate naturally with machines as they do with their peers. The existing approaches that use body language for teaching machines often rely on supervised learning approaches and on the system designer to map these cues to instructions or rewards.

These supervised approaches require labeled datasets for recognizing the different non-verbal cues produced by the people. However, it is a difficult, cumbersome and time-consuming process to create an extensive labeled dataset for recognizing nuanced non-verbal cues, that are produced by people in a variety of situations.

The existing approaches to human-machine interaction map body language to instructions or rewards, assuming that these non-verbal cues have similar meaning across different people. The body language of people differs with their situations and preferences. In such cases, the machines requires those users' to produce unnatural cues in order to effectively teach them. Similar unintuitive process occurs in approaches that use body language as instructions to the

machine.

Ideally, the meaning of these cues needs to be learned from ongoing interactions by the machine as different cues can have different meanings and these depend on the people who produce them.

In this thesis, we introduced an approach that does not rely on large supervised learning datasets for teaching machines through body language. More importantly, our approach learns the meaning of these cues as they are produced by the user during ongoing interactions, thereby alleviating this problem of creating an extensive dataset or hand-engineering their meaning to the machine. By learning the meaning of these cues, the users can teach their machines to behave according to their preferences. We would like to point out here that this is a significant contribution and is vastly different from conventional human-machine interaction approaches.

7.2 How Does Our Approach Differ from Approaches that Learn from User-Generated Rewards?

In the previous section, we discussed the significance of our prospective body language approach over existing human-machine interaction approaches that use body language. In this section, we draw out the important differences between ours and those that learn from user-generated rewards.

Many approaches, like the ones introduced by Knox and Stone (2009) and by Iturrate et al. (2015), were developed recently for improving human-machine interaction through reinforcement learning. These techniques allow the user to directly manipulate the existing reward channel through user-generated rewards, thereby teaching the machine to act according to their preferences. In these approaches, the user generates rewards, which complements the reward signal from the environment. Using this combined reward signal, the agent learns and adapts its behavior in order to maximize its rewards.

These approaches to human-machine interaction requires constant super-

vision from the user in the form of user-generated rewards throughout the operation of the machine. This contradicts the objective of human-machine interaction, which is that the agent needs to complete a task according to their user preferences without involving much effort from the user. More specifically, these interaction techniques should not require constant supervision from their users. In order to meet this requirement, the machines needs to have some level of autonomy and not expect expensive feedback throughout the course of its operation.

A simple way of meeting these requirements is by adding the user's body language into the existing state space, which is exactly what our approach does. As a result, the agent learns to predict its future rewards directly from the user's body language, which is a powerful source of feedback compared to user-generated rewards. More importantly, these body language cues are produced without much effort as opposed to the process of producing user-generated rewards.

7.3 Do People Produce Non-Verbal Cues While Interacting With Their Machines?

In the previous chapters, we introduced our approach, where users can teach their machines using body language cues without involving much effort from the users. As opposed to existing methods, our approach allows people to use high-bandwidth channels for communicating feedback to their machines. However, in our experiments, we assumed that the users produced facial expression cues during their interactions with their agents. Naturally, this brings up the following question: do people produce facial expressions or other body language cues while interacting with their agents?

People often produce subtle body language cues while communicating with their peers. Many research works in psychology and affective neuroscience (Adolphs, 2002; De Gelder, 2006; Whalen et al., 2013; Jack & Schyns, 2015; Scheider et al. 2016) have studied this specific phenomenon. More importantly, they study why people produce such non-verbal cues and how they

influence their observers.

It is reasonable to think that people do not produce such subtle cues while interacting with machines. However, recently, Fridman et al. (2017) discovered that people produce significant body language while interacting with their machines. Specifically, their work studied the predictive power of various body language cues produced naturally by a user while driving a semi-autonomous car through different environments. More importantly, their results suggest that these non-verbal cues were powerful enough to predict different aspects of their environments and about the car, like the traffic density, cruise control, road conditions, weather, and proximity to an intersection among others. They created a dataset with 100 users logging in 2 million vehicle miles, which resulted in $43,000$ hours of supervised training data. The entire process of creating this dataset and subsequently labeling this data took about 13 months.

Another research work, by Abdić et al. (2016), studied how useful these body language cues were for predicting their satisfaction. They used a dataset obtained from a user study, which involved the users to drive a car and operate a voice-based navigation system. The users' faces along with their conversation with the navigation system were recorded and labeled with their satisfaction level. The authors trained a supervised learning approach over this dataset and found that their approach predicted the user satisfaction with significant accuracy. More importantly, they reported that the visual stream containing the driver's face was more predictive of their satisfaction than their conversation with the navigation system.

From these research works, it is clear that people do produce natural non-verbal cues while interacting with their machines. More recently, Fridman et al. (2017a) and Fridman et al. (2017b) are beginning to build upon these results, by designing better user-interfaces for semi-autonomous vehicles.

7.4 How Does Our Approach Relate to Affective Computing?

Affective computing is defined as the study and development of systems that can recognize, interpret, process and simulate human affects, where an affect is the experience of a feeling or emotion. Sometimes, affect is also defined as facial, vocal or gesture behavior that serves as a response to a stimuli. The motivation of affective computing is to simulate empathy in computers. In short, affective computing machines need to interpret the state of mind of people and adapt its behavior towards them. This modern field was formalized by Rosalind Picard in the seminal paper published in 1995.

From the definitions of affective computing, it is clear that our prospective body language approach fits these requirements. Specifically, our approach seems to be the initial steps towards building a complete system that identifies the user's state of mind from their body language and subsequently adapts their behavior towards them. It is also important to point out here that the existing research works in affective computing rely on supervised learning approaches for recognizing different affects from the user. It is surprising to see that our work is the first to comprehensively address this affective computing problem, without making any assumptions about the user's body language. Also, reinforcement learning seems to be promising approach for affective computing.

7.5 Better Feature Representations for Nuanced Body Language

The most important aspect that needs to be improved upon in our approach would be the feature representation that forms the state representation of the agent.

In the experiments described in this thesis, we processed facial landmarks of the interacting user, which was extracted at each time step. These facial landmarks formed the feature representation to the agent. Such a feature

representation is not powerful enough to encode cues that extend for many time steps, like nodding the head or curling the lips. People often produce such nuanced body language cues, which contain significant communicative value to the agent. By using better feature representations, the agent can learn from such nuanced cues, and as a result, produce better user experience.

A better approach towards building powerful feature representation would involve the recent advances made in the field of representation learning. In recent times, the trend is to learn features directly from the data by minimizing some objective function. These representation learning approaches involves constructing a hierarchical network structure, called a neural network, whose parameters are tuned through gradient-based optimization algorithms to solve a given task. This hierarchical organization enables these networks to learn nonlinear feature representations directly from the data, substantially improving its performance.

This approach of learning hierarchical representations grounded directly in the data has achieved significant state of the art results in many domains, like translating languages (Cho et al., 2014), predicting patient survival (Miotto et al., 2016), playing poker (Moravčík et al., 2017), Go (Silver et al., 2016) and atari (Mnih et al., 2015). The performance improvements obtained through these hierarchical representation learning methods are many-folds over conventional hand-engineered methods.

We hypothesize that, by using neural networks, our approach can learn temporally extended and nuanced body language cues produced by the user, resulting in more powerful interaction techniques between people and their machines.

7.6 Integrating Different Modalities

Our approach of teaching a machine using prospective body language was motivated by the importance of improving human-machine interaction by studying human communication. However, in order to achieve its complete potential, we need to integrate this with speech and other body language.

People produce many different body language that range across multiple modalities and they usually differ with people. For example, people tend to accentuate their hands while they speak. Sometimes, the posture exhibited by people also have valuable information about their state of mind. Learning to understand such body language allows machines to better understand their user and their preferences. Integrating different modalities would allow the machines to better learn about their users and will help the research community to significantly progress towards the ambitious goal of intelligence amplification.

Chapter 8

Conclusion

Body language plays a key role in human communication. These non-verbal cues complement the speech in human communication and simplify the process of communication with our peers. More importantly, these cues are produced naturally during the process of communication and usually have communicative value. Our peers on observing these cues subsequently adapt their behavior towards us.

An important goal of artificial intelligence is to amplify our cognitive and physical capabilities of people through machines. This is called intelligence amplification. As a first step towards achieving this grand prize, we need to design sustainable forms of human-machine interaction that closely adheres to the principles underlying human communication. Specifically, we need to design approaches that allow humans to communicate naturally with their machines, similar to how people communicate with each other.

Many of the existing approaches to human-machine interaction that use body language assume that the meaning of different cues are given to the machine from some external source. This is not a sound assumption to make because these cues usually vary with people. As a result, these existing approaches requires the user to actively translate their non-verbal cues in order to fit the system design. This process of translation required from the user is unnatural and causes cognitive load on the user, making these existing approaches unsustainable. More importantly, these existing approaches do not address the goal of this thesis — to create approaches that learn from people

without their really trying.

In this thesis, we introduced our prospective body language approach, which allows people to teach their machines using whatever body language cues they produce during the process of interaction. More specifically, these cues indicate their satisfaction towards the machine's behavior. First, the agent learns to predict its future rewards by mapping the user's body language with the occasionally generated explicit feedback. This mapping is achieved through a value function. Second, after this value function reliably predicts its future rewards from the cues, the agent perceives these cues as useful evaluative feedback signals, informing the agent about its performance. More importantly, the agent utilizes this feedback to adapt its behavior so as to maximize its rewards and user satisfaction. This process of influencing behavior through subtle cues is similar to how people adapt their behaviors in human communication.

As opposed to the existing approaches, our method does not require any supervised learning datasets for recognizing the subtle cues produced by the user. Crucially, our method does not assume anything regarding the user's body language as our method learns the meaning of these cues through ongoing interactions with the user. As a result, our approach does not require the user to translate their cues according to the system design. We believe this to be the first work in directly addressing the overall goal of this thesis, where an implemented system learns from people without involving much effort from the people.

We expect that the our prospective body language approach can be easily extended to many real-world human-machine interactions domains, like intelligent prostheses, and self-driving cars. In the near future, we expect our general human-machine interaction approach to improve the quality of life for people by amplifying or augmenting their physical and cognitive capabilities through machines.

Bibliography

Abbeel, P., Coates, A., Quigley, M., & Ng, A. Y. (2007). An application of reinforcement learning to aerobatic helicopter flight. In *Advances in neural information processing systems* (pp. 1-8).

Abbeel, P., & Ng, A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning* (p. 1). ACM.

Abdić, I., Fridman, L., McDuff, D., Marchi, E., Reimer, B., & Schuller, B. (2016). Driver frustration detection from audio and video in the wild. In *KI 2016: Advances in Artificial Intelligence: 39th Annual German Conference on AI, Klagenfurt, Austria, September 26-30, 2016, Proceedings* (Vol. 9904, p. 237). Springer.

Adolphs, R. (2002). Trust in the brain. *Nature neuroscience*, 5(3), 192-193.

Alizadeh, T., Calinon, S., & Caldwell, D. G. (2014). Learning from demonstrations with partially observable task parameters. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (pp. 3309-3314). IEEE.

Argall, B., Browning, B., & Veloso, M. (2007). Learning by demonstration with critique from a human teacher. In *Human-Robot Interaction (HRI), 2007 2nd ACM/IEEE International Conference on* (pp. 57-64). IEEE.

Barto, A. G., Sutton, R. S., & Anderson, C. W. (1983). Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE transactions on systems, man, and cybernetics*, (5), 834-846.

Bhatnagar, S., Sutton, R., Ghavamzadeh, M., & Lee, M. (2009). Natural actor-critic algorithms. *Automatica*, 45(11).

Breazeal, C. (1998). Regulating human-robot interaction using 'emotions','drives', and facial expressions. *In Proceedings of Autonomous Agents* (Vol. 98, pp. 14-21).

Breazeal, C. L. (2004). *Designing sociable robots*. MIT press.

Breazeal, C. (2009). Role of expressive behaviour for robots that learn from people. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 364(1535), 3527-3538.

Cakmak, M., & Thomaz, A. L. (2012). Designing robot learners that ask good questions. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction* (pp. 17-24). ACM.

Chen, D. L., & Mooney, R. J. (2011). Learning to Interpret Natural Language Navigation Instructions from Observations. In *AAAI* (Vol. 2, pp. 1-2).

Chernova, S., & Veloso, M. (2008). Teaching collaborative multi-robot tasks through demonstration. In *Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on* (pp. 385-390). IEEE.

Cobo, L. C., Isbell Jr, C. L., & Thomaz, A. L. (2012). Automatic task decomposition and state abstraction from demonstration. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1* (pp. 483-490). International Foundation for Autonomous Agents and Multiagent Systems.

Cruz, F., Twiefel, J., Magg, S., Weber, C., & Wermter, S. (2015). Interactive reinforcement learning through speech guidance in a domestic scenario. In *Neural Networks (IJCNN), 2015 International Joint Conference on* (pp. 1-8). IEEE.

Darwin, C. (1998). *The expression of the emotions in man and animals*. Oxford University Press, USA.

De Gelder, B. (2006). Towards the neurobiology of emotional body language. *Nature reviews. Neuroscience*, 7(3), 242.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Faria, B. M., Reis, L. P., & Lau, N. (2012). Cerebral palsy eeg signals classification: Facial expressions and thoughts for driving an intelligent wheelchair. In *Data Mining Workshops (ICDMW), 2012 IEEE 12th International Conference on* (pp. 33-40). IEEE.

Faria, P. M., Braga, R. A., Valgode, E., & Reis, L. P. (2007). Interface framework to drive an intelligent wheelchair using facial expressions. In *Industrial Electronics, 2007. ISIE 2007. IEEE International Symposium on* (pp. 1791-1796). IEEE.

Fridman, Toyoda, Seaman, Seppelt, Angell, Lee, Mehler. (2017). What Can Be Predicted from 6 Seconds of Driver Glances?. *CHI.*

Fridman, Reimer, Mehler, Freeman. (2017). Cognitive Load Estimation in a Large On-Road Driving Dataset. (under review)

Ferreira, A., Silva, R. L., Celeste, W. C., Bastos Filho, T. F., & Sarcinelli Filho, M. (2007). Human–machine interface based on muscular and brain signals applied to a robotic wheelchair. In *Journal of Physics: Conference Series* (Vol. 90, No. 1, p. 012094). IOP Publishing.

Galán, F., Nuttin, M., Lew, E., Ferrez, P. W., Vanacker, G., Philips, J., & Millán, J. D. R. (2008). A brain-actuated wheelchair: asynchronous and non-invasive brain–computer interfaces for continuous control of robots. *Clinical Neurophysiology*, 119(9), 2159-2169.

Graham, J. A., & Argyle, M. (1975). A cross-cultural study of the communication of extra-verbal meaning by gestures. *International Journal of Psychology*, 10(1), 57-67.

Hollender, N., Hofmann, C., Deneke, M., & Schmitz, B. (2010). Integrating cognitive load theory and concepts of human–computer interaction. *Computers in Human Behavior*, 26(6), 1278-1288.

Isbell, C. L., Kearns, M., Kormann, D., Singh, S., & Stone, P. (2000). Cobot in LambdaMOO: A social statistics agent. In *AAAI/IAAI* (pp. 36-41).

Iturrate, I., Chavarriaga, R., Montesano, L., Minguez, J., & Millán, J. D. R. (2015). Teaching brain-machine interfaces as an alternative paradigm to neuroprosthetics control. *Scientific reports*, 5, 13893.

Iturrate, I., Montesano, L., & Minguez, J. (2010). Robot reinforcement learning using EEG-based reward signals. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on* (pp. 4822-4829). IEEE.

Jack, R. E., & Schyns, P. G. (2015). The human face as a dynamic tool for social communication. *Current Biology*, 25(14), R621-R634.

Judah, K., Roy, S., Fern, A., & Dietterich, T. G. (2010). Reinforcement Learning Via Practice and Critique Advice. In *AAAI*.

Kazemi, V., & Sullivan, J. (2014). One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1867-1874).

Kaochar, T., Peralta, R., Morrison, C., Fasel, I., Walsh, T., & Cohen, P. (2011). Towards understanding how humans teach robots. *User modeling, adaption and personalization*, 347-352.

Kim, E. S., & Scassellati, B. (2007). Learning to refine behavior using prosodic feedback. In *Development and Learning, 2007. ICDL 2007. IEEE 6th International Conference on* (pp. 205-210). IEEE.

Knox, W. B., & Stone, P. (2009). Interactively shaping agents via human reinforcement: The TAMER framework. In *Proceedings of the fifth international conference on Knowledge capture* (pp. 9-16). ACM.

Knox, W. B., & Stone, P. (2015). Framing reinforcement learning from human reward: Reward positivity, temporal discounting, episodicity, and performance. *Artificial Intelligence*, 225, 24-50.

Knox, W. B., Glass, B. D., Love, B. C., Maddox, W. T., & Stone, P. (2012). How humans teach agents. *International Journal of Social Robotics*, 4(4), 409-421.

Koenig, N. P., & Mataric, M. J. (2012). Training Wheels for the Robot: Learning from Demonstration Using Simulation. In *AAAI Fall Symposium: Robots Learning Interactively from Human Teachers.*

Konda, V. R., & Tsitsiklis, J. N. (2000). Actor-critic algorithms. In *Advances in neural information processing systems* (pp. 1008-1014).

Lievesley, R., Wozencroft, M., & Ewins, D. (2011). The Emotiv EPOC neuroheadset: an inexpensive method of controlling assistive technologies using facial expressions and thoughts?. *Journal of Assistive Technologies*, 5(2), 67-82.

Liu, K., & Picard, R. W. (2003). Subtle expressivity in a robotic computer. In *CHI 2003 Workshop on Subtle Expressiveness in Characters and Robots.*

Mathewson, K. W., & Pilarski, P. M. (2017). Reinforcement Learning based Embodied Agents Modelling Human Users Through Interaction and Multi-Sensory Perception. *arXiv preprint arXiv:1701.02369.*

Mehler, B., Reimer, B., Dobres, J., Foley, J., & Ebe, K. (2016). *Additional Findings on the Multi-Modal Demands of "Voice-Command" Interfaces* (No. 2016-01-1428). SAE Technical Paper.

Miotto, R., Li, L., Kidd, B. A., & Dudley, J. T. (2016). Deep patient: An unsupervised representation to predict the future of patients from the electronic health records. *Scientific reports*, 6, 26094.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G. and Petersen, S. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529-533.

Moravčík, M., Schmid, M., Burch, N., Lisý, V., Morrill, D., Bard, N., Davis, T., Waugh, K., Johanson, M. and Bowling, M. (2017). Deepstack: Expert-level artificial intelligence in no-limit poker. *arXiv preprint arXiv:1701.01724.*

Pezzulo, G., Donnarumma, F., & Dindo, H. (2013). Human sensorimotor communication: A theory of signaling in online social interactions. *PloS one*, 8(11), e79876.

Pfungst, O. (1911). *Clever Hans:(the horse of Mr. Von Osten.) a contribution to experimental animal and human psychology.* Holt, Rinehart and Winston.

Pilarski, P. M., Dawson, M. R., Degris, T., Fahimi, F., Carey, J. P., & Sutton, R. S. (2011). Online human training of a myoelectric prosthesis controller via actor-critic reinforcement learning. In *Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on* (pp. 1-7). IEEE.

Pilarski, P. M., Sutton, R. S., & Mathewson, K. W. (2015). Prosthetic Devices as Goal-Seeking Agents. In *2nd Workshop on Present and Future of Non-Invasive Peripheral-Nervous-System Machine Interfaces*, Singapore.

Rechy-Ramirez, E. J., Hu, H., & McDonald-Maier, K. (2012). Head movements based control of an intelligent wheelchair in an indoor environment. In *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on* (pp. 1464-1469). IEEE.

Reis, L. P., Braga, R. A., Sousa, M., & Moreira, A. P. (2009). IntellWheels MMI: A flexible interface for an intelligent wheelchair. In *Robot Soccer World Cup* (pp. 296-307). Springer, Berlin, Heidelberg.

Rosselli, M., & Ardila, A. (2003). The impact of culture and education on non-verbal neuropsychological measurements: A critical review. *Brain and cognition*, 52(3), 326-333.

Schulman, J., Ho, J., Lee, C., & Abbeel, P. (2013). Generalization in robotic manipulation through the use of non-rigid registration. In *Proceedings of the 16th International Symposium on Robotics Research* (ISRR).

Scheider, L., Waller, B. M., Oña, L., Burrows, A. M., & Liebal, K. (2016). Social use of facial expressions in hylobatids. *PloS one*, 11(3), e0151733.

She, L., Cheng, Y., Chai, J. Y., Jia, Y., Yang, S., & Xi, N. (2014). Teaching robots new actions through natural language instructions. In *Robot and Human Interactive Communication, 2014 RO-MAN: The 23rd IEEE International Symposium on* (pp. 868-873). IEEE.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ⋯ & Dieleman, S. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587), 484-489.

Suay, H. B., & Chernova, S. (2011). Effect of human guidance and state space size on interactive reinforcement learning. In *RO-MAN, 2011 IEEE* (pp. 1-6). IEEE.

Sutton, R. S., & Barto, A. G. (2017). Reinforcement learning: An introduction. (in press).

Sutton, R. S., Mahmood, A. R., & White, M. (2016). An emphatic approach to the problem of off-policy temporal-difference learning. *The Journal of Machine Learning Research*, 17(1), 2603-2631.

Tanaka, K., Matsunaga, K., & Wang, H. O. (2005). Electroencephalogram-based control of an electric wheelchair. *IEEE transactions on robotics*, 21(4), 762-766.

Thomaz, A. L., & Breazeal, C. (2006). Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance. In *AAAI* (Vol. 6, pp. 1000-1005).

Thomaz, A. L., & Breazeal, C. (2008). Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence*, 172(6-7), 716-737.

van Seijen, H., Mahmood, A. R., Pilarski, P. M., Machado, M. C., & Sutton, R. S. (2016). True online temporal-difference learning. *Journal of Machine Learning Research*, 17(145), 1-40.

van Seijen, H. (2016). Effective multi-step temporal-difference learning for non-linear function approximation. *arXiv preprint arXiv:1608.05151.*

Vasan, G., & Pilarski, P. M. (2017) Learning from Demonstration: Teaching a Myoelectric Prosthesis with an Intact Limb via Reinforcement Learning. In *Rehabilitation Robotics (ICORR), 2017 IEEE International Conference on*. IEEE.

Whalen, P. J., Raila, H., Bennett, R., Mattek, A., Brown, A., Taylor, J., & Palmer, A. (2013). Neuroscience and facial expressions of emotion: The role of amygdala–prefrontal interactions. *Emotion Review*, 5(1), 78-83.

Williams III, T. W. (2011). Progress on stabilizing and controlling powered upper-limb prostheses. *Journal of Rehabilitation Research & Development*, 48(6), ix-ix.

Yammiyavar, P., Clemmensen, T., & Kumar, J. (2008). Influence of cultural background on non-verbal communication in a usability testing situation. *International Journal of Design*, 2(2).