Affect and Learning: a computational analysis

Broekens, D.J.

6

Affect as Reinforcement

Affective Expressions Facilitate Robot Learning

Until now, we have used affect as an abstraction for how well the agent is doing. This abstraction is a long-term signal, based on the reinforcement the agent receives during the process of learning. We have experimented with controlling meta-parameters by means of artificial affect (exploration versus exploitation, Chapter 3; broad versus narrow thoughts implemented via internal simulation of potential interaction with the grid world, Chapter 4). Artificial affect was thus related to mood (long timescale, not directed at a specific situation). Furthermore, artificial affect was a signal originating from the agent itself.

Affect can also be an abstraction for the positiveness versus negativeness of a *current* situation or object, as well as being elicited more directly by an external source (e.g., when used in affective communication). In this chapter we take such an approach. We thus part from the definition of affect introduced in Chapter 2. In this chapter, affect is a short-term signal communicated by a human observer to a learning simulated robot. The common part in this definition of affect and the one introduced in Chapter 2 is that affect still is an abstraction for positive versus negative.

In this chapter we briefly present *EARL*, our framework for the systematic study of the relation between *e*motion, *a*daptation and *r*einforcement *l*earning. *EARL* is a framework, currently a prototype, that embodies many of the ways in which affect can influence learning, when learning is conceptualized as Reinforcement Learning (RL). *EARL* enables the study of, among other things, (a) affect as reinforcement to the robot (both internally generated as well as socially communicated; this chapter), (b) affect as perceptual feature to the robot (again internally generated and social), (c) affect resulting from reinforced robot behavior (see also Chapter 2), and (d) affect as meta-parameters for the robot's learning mechanism (Chapter 3 and 4). *EARL* can be seen as the concretization of the insights developed while researching the topics described in Chapters 2 to 5 of this thesis.

In this chapter, we focus on one aspect of *EARL*: the ability to model communicated affect by a human observer used as reinforcement by the robot. In humans, emotions are crucial to learning. For example, a parent—observing a child—uses emotional expression to encourage or discourage specific behaviors. Emotional expression can therefore be a reinforcement signal to a child. We hypothesize that affective facial expressions facilitate robot learning, and compare a *social* setting with a *non-social* one to test this. The non-social setting consists

of a simulated robot that learns to solve a typical RL task in a continuous grid-world environment. The social setting additionally consists of a human (parent) observing the simulated robot (child). The human's emotional expressions are analyzed in real time and converted to an additional reinforcement signal used by the robot; positive expressions result in reward, negative expressions in punishment. We quantitatively show that the "social robot" indeed learns to solve its task significantly faster than its "non-social sibling". We conclude that this presents strong evidence for the potential benefit of affective communication with humans in the Reinforcement Learning loop.

6.1 Introduction

In humans, emotion influences thought and behavior in many ways (Custers & Aarts, 2005; Damasio, 1994; Dreisbach & Goschke, 2004; Rolls, 1999). For example, emotion influences how humans process information by controlling the broadness versus the narrowness of attention (see also Chapter 3 and 4). Also, emotion functions as a social signal that communicates reinforcement of behavior in, e.g., parent-child relations. Computational modeling (including robot modeling) has proven to be a viable method of investigating the relation between emotion and learning (Broekens, Kosters & Verbeek, 2007; Gadanho, 2003), emotion and problem solving (Belavkin, 2004; Bothello & Coehlo, 1998), emotion and social robots (Breazeal, 2001; for review see Fong, Nourbakhsh & Dautenhahn, 2003), and emotion, motivation and behavior selection (Avila-Garcia & Cañamero, 2004; Blanchard and Cañamero, 2006; Cos-Aguilera et al., 2005; Velasquez, 1998). Although many approaches exist and much work has been done on computational modeling of emotional influences on thought and behavior, none explicitly targets the study of the relation between emotion and learning using a complete end-to-end framework in a Reinforcement Learning context[1]. By this we mean a framework that enables systematic *quantitative* study of the relation between affect and RL in a large variety of ways, including (a) affect as reinforcement to the robot (both internally generated as well as socially communicated), (b) affect as perceptual feature to the robot (again internally generated and social), (c) affect resulting from reinforced robot behavior, and (d) affect as meta-parameters for the robot's learning mechanism. In this chapter we present such a framework. We call our framework *EARL*, short for the systematic study of the relation between *e*motion, *a*daptation and *r*einforcement *l*earning.

[1] Although the work by Gadanho (2003) is a partial exception as it explicitly addresses emotion in the context of RL. However, this work does not address social human input and social robot output.

Here we specifically focus on the influence of socially communicated emotion on learning in a Reinforcement Learning context. We show, using our framework *EARL*, that human emotional expressions can be used as an additional reinforcement signal used by a simulated robot.

The robot's task is to optimize food-finding behavior while navigating through a continuous grid-world environment. The grid world is not discrete, nor is an attempt made to define discrete states based on the continuous input. The grid world contains walls, path and food patches. The robot perceives its direct surroundings as they are. We have developed an action-based learning mechanism that learns to predict values of actions based on the current perception of the agent (note that in this chapter we use the terms agent and robot interchangeably). Every action has its own Multi-Layer Percepton network (see also Lin, 1993) that learns to predict a modified version of the *Q*-value (Sutton & Barto, 1998). We have used this setup so that observed robot behavior can be extrapolated to the real world; building the actual robot with appropriate sensors and actuators would, in theory, suffice to replicate the results. We explain our modeling method in more detail in Section 6.5.

As mentioned above, we study the effect of a human's emotional expression on the learning behavior of the robot. In humans, emotions are crucial to learning. For example, a parent—observing a child—uses emotional expression to encourage or discourage specific behaviors. In this case, the emotional expression is used to setup an *affective communication channel* (Picard, 1997) and is used to communicate a reinforcement signal to a child. In this chapter we take *affect* to mean the positiveness versus the negativeness of a situation, object, etc. (see Rolls, 1999; Russell, 2003; and Broekens, Kosters & Verbeek, 2007, or Chapter 2 for a more detailed argumentation of this point of view). The human observes the simulated robot while it learns to find food, and affect in the human's facial expression is recognized by the robot in real time[2]. A smile is interpreted as communicating positive affect and therefore converted to a small additional reward (additional to the reinforcement the robot receives from its simulated environment). The expression of fear is interpreted as communicating negative affect and therefore converted to a small additional punishment. We call this the *social* setting. The non-social setting is a standard experimental Reinforcement Learning setup without human input.

[2] In this chapter, affect is thus a short-term signal elicited by an external source, as opposed to affect defined in Chapter 2 where it is a long-term signal elicited by mechanisms in the agent itself based on its learning performance.

We hypothesized that robot learning (in a RL context as described above) is facilitated by additional social reinforcement. Our experimental results support this hypothesis. We compared the learning performance of our simulated robot in the social and non-social settings, by analyzing averages of learning curves. The main contribution of this research is that it presents *quantitative* evidence of the fact that a human-in-the-loop can boost learning performance in real-time, in a plausible learning environment. We belief this is an important result. It provides a solid base for further study of human mediated robot learning in the context of real-world applicable Reinforcement Learning, using the communication protocol nature has provide for that purpose, i.e., emotional expression and recognition. Therefore, our results suggest that robots can be trained and their behaviors optimized using natural social cues. This facilitates human-robot interaction.

The rest of this chapter is structured as follows. In Section 6.2 we explain in some more detail our view of affect, emotion and how affect influences learning in humans. In Section 6.3 we briefly introduce *EARL*, our complete framework. In Section 6.4 we describe how communicated affect is linked to a social reinforcement signal. In Section 6.5, we explain our method of study (e.g., the grid world, the learning mechanism). Section 6.6 discusses the results and Section 6.7 discusses these in a broader context and presents concluding remarks and future work.

6.2 Affect as Reinforcement

As we have seen in the previous chapters, affect influences thought and behavior in a variety of ways. For example, a person's mood influences processing style and attention, emotions influence how one thinks about objects, situations and persons, and emotion is related to learning behaviors as well as can be used to modify learning parameters in artificial learning agents. So, affect regulates behavior.

Affect also regulates behavior of others. Obvious in human development, expression (and subsequent recognition) of emotion is important to communicate (dis)approval of the actions of others. This is typically important in parent-child relations. Parents use emotional expression to guide behavior of infants. Emotional interaction is essential for learning. Striking examples are children with an autistic spectrum disorder, typically characterized by a restricted repertoire of behaviors and interests, as well as social and communicative impairments such as difficulty in joint attention, difficulty recognizing and expressing emotion, and lacking of a social smile (for review see Charman & Baird, 2002). Apparently, children suffering from this disorder have both a

difficulty in building up a large set of complex behaviors *and* a difficulty understanding emotional expressions and giving the correct social responses to these. This disorder provides a clear example of the interplay between learning behaviors and being able to process emotional cues.

In this chapter we specifically focus on the influence of socially communicated affect on learning: we focus on the role of affect in guiding learning in a social human-robot setting. We use affect to denote the positiveness versus negativeness of a situation. We ignore the arousal a certain situation might bring. Positive affect characterizes a situation as good, while negative affect characterizes that situation as bad (e.g., Russell, 2003). Further, we use affect to refer to the *short term* timescale: i.e., to emotion. We hypothesize that affect communicated by a human observer can enhance robot learning. In our study we assume that the recognition of affect translates into a reinforcement signal. Thus, the robot uses a *social reinforcement* in addition to the reinforcement it receives from its environment while it is building a model of the environment using Reinforcement Learning mechanisms. In the following sections we first explain our framework after which we detail our method and discuss results and further work.

6.3 *EARL*: A Computational Framework to Study the Relation between Emotion, Adaptation and Reinforcement Learning.

To study the relation between emotion, adaptation and Reinforcement Learning, we have developed an end-to-end framework. The framework consists of four parts:

- An emotion recognition module, recognizing emotional facial expression in real time.
- A Reinforcement Learning agent to which the recognized emotion can be fed as input.
- An artificial emotion module slot, this slot can be used to plug in different models of emotion into the learning agent that produce the artificial emotion of the agent as output. The modules can use all of the information that is available to the agent (such as action repertoire, reward history, etc.). This emotion can be used by the agent as intrinsic reward, as metalearning parameter, or as input for the expression module.
- An expression module, consisting of a robot head with the following degrees of freedom: eyes moving up and down, ears moving up and down on the outside, lips moving up and down, eyelids moving up and down on the outside, and RGB eye colors.

Emotion recognition is based on quite a crude mechanism using the face tracking abilities of OpenCV[3]. It uses 9 points on the face each defined by a blue sticker: 1 on the tip of the nose, 2 above each eyebrow, 1 at each mouth corner and 1 on the upper and lower lip. The recognition module is configured to store multiple prototype point constellations. The user is prompted to express a certain emotion and press space while doing so. For every emotional expression (in the case of our experiment neutral, happy and afraid), the module records the positions of the 9 points relative to the nose. This is a prototype point vector. After configuration, to determine the current emotional expression in real time, the module calculates a weighted distance from the current point vector (read in real-time from a web-cam mounted on the computer screen) to the prototype vectors. Different points get different weights. This results in an error measure for every prototype expression. This error measure is the basis for a normalized vector of recognized emotion intensities. The recognition module sends this vector to the agent (e.g., neutral 0.3, happy 0.6, fear 0.1). Our choice of weights and features has been inspired by work of others (for review see Pantic & Rothkrantz, 2000). Of course the state of the art in emotion recognition is more advanced than our current approach. However, as our focus is affective learning and not the recognition process per se, we contented ourselves with a low fidelity solution (working almost perfectly for neutral, happy and afraid, when the user keeps the head in about the same position).

Note that we do not aim at generically recognizing emotional expressions. Instead, we tune the recognition module to the individual observer to accommodate his/her personal and natural facial expressions.

The Reinforcement Learning agent receives this recognized emotion and can use this in multiple ways: as reward, as information (additional state input), as metaparameter (e.g., to control learning rate), and as social input directly into its emotion model. In this chapter we focus on social reinforcement, in particular on the recognized emotion being used as additional reward or punishment. The agent, its learning mechanism and how it uses the recognized emotion as reinforcement are detailed in Sections 6.4 and 6.5.

The artificial emotion model slot enables us to plug in different emotion models based on different theories to study their behavior in the context of Reinforcement Learning. For example, we have developed a model based on the theory by Rolls (1999), who argues that many emotions can be related to reward and punishment and the lack thereof. This model enables us to see if the agent's situation results in a plausible (e.g., scored by a set of human observers) emotion

[3] http://www.intel.com/technology/computing/opencv/index.htm

emerging from the model. By scoring the plausibility of the resulting emotion, we can learn about the compatibility of, e.g., Rolls' emotion theory with Reinforcement Learning. However, in the current study we have not used this module, as we focus on affective input as social reward.

The emotion expression part is a physical robot head. The head can express an arbitrary emotion by mapping it to its facial features, again according to a certain theory. Currently our head expresses emotions according to the Pleasure Arousal Dominance (PAD) model by Mehrabian (1980). We have a continuous mapping from the 3-dimensional PAD space to the features of the robot face. As such we do not need to explicitly work with emotional categories or intensities of the categories. The mapping appears to work quite well, but is in need of validation study (again using human observers). We have not used the robot head for the studies reported upon in this chapter.

We now describe in detail how we coupled the recognized human emotion to the social reinforcement signal for the robot. Then we explain in detail our adapted Reinforcement Learning mechanism (such that it enabled learning in continuous environments), and our method of study as well as our results.

6.4 Emotional Expressions as Reinforcement Signal.

As mentioned earlier, emotional expressions and facial expressions in particular can be used as social cues for the desirability of a certain action. In other words, an emotional expression can express reward and punishment if directed at an individual. We focus on communicated affect, i.e., the positiveness versus negativeness of the expression. If the human expresses a smile (happy face) this is interpreted as positive affect. If the human expresses fear, this is interpreted as negative affect. We interpret a neutral face as affectless.

We have studied the mechanism of communicated affective feedback in a human-robot interaction setup. The human's face is analyzed (as explained above) and a vector of emotional expression intensities is fed to the learning agent. The agent takes the expression with the highest intensity as dominant, and equates this with a *social reward* of, e.g., 2 (happy), −2 (fear) and 0 (neutral). This is obviously a simplified setup, as the human face communicates much more subtle affective messages and at the very least is able to communicate the degree of reward and punishment. However, to investigate our hypothesis (affective human feedback increases robot learning performance), the just described mechanism is sufficient.

The social reward is simply added to the "normal" reward the agent receives from the environment. So, if the agent walks on a path somewhere in the grid world, it receives a reward (say 0), but when the user smiles, the resulting actual reward becomes 2, while if the user looks afraid, the resulting reward becomes −2.

6.5 Method

To study the impact of social reinforcement on robot learning, we have used our framework in the following experimental setup.

A simulated robot (agent) "lives" in a continuous grid-world environment consisting of wall, food and path patches (Figure 6.1). These are the features of the world observable by the agent. The agent cannot walk on walls, but can walk on path and food. Walls and path are neutral (have a reinforcement of 0.0), while food has a reinforcement of 10. One cell in the grid is assumed to be a 20 by 20 spatial unit object (let's say 20 x 20 centimeters). Even though wall, path and food are placed on a grid, the world is continuous in the following sense: the agent moves by turning or walking in a certain direction using an arbitrary speed (in our experiments set at 3 spatial units per time unit), and perceives its direct surroundings (within a radius of 20 spatial units) according to its looking direction (one out of 16 possible directions).

The agent uses a "relative eight-neighbor metric" meaning that it perceives features of the world at 8 points around it, with each point at a distance of 20 from the center point of the agent and each point at an interval of 1/4 PI radians, with the first point always being exactly in front of it (Figure 6.1).

The state perceived by the agent (its percept) is a real-valued vector of inputs between 0 and 1; each input is defined by the relative contribution of a certain feature in the agent-relative direction corresponding to the input. For example, if the agent sees a wall just in front of it (i.e., the center point of a wall object is exactly at a distance of 20 units as measured from the current agent location in its looking direction) the first value in its perceived state would be equal to 1. This value can be anywhere between 0 and 1 depending on the distance of that point to the feature. For the three types of features, the agent thus has 3x8=24 real-valued inputs between 0 and 1 as its perceived world state *s* (Figure 6.1). Therefore the agent can approach objects (e.g., a wall) from a large number of possible angles and positions, with every intermediate position being possible.

For all practical purposes, the learning environment can be considered continuous. States are not discretized to facilitate learning. Instead we chose to

use the perceived state as is, to maximize compatibility of our experimental results with real-world robots. However, Reinforcement Learning in continuous environments introduces several important problems for standard RL techniques, such as *Q*-learning, mainly because a large number of potentially similar states exist as well as a very long path length between start and goal states can occur making value propagation difficult.

We now briefly explain our adapted RL mechanism. As RL in continuous environments is not specifically the topic of the chapter we have left out some of the rational for our choices.

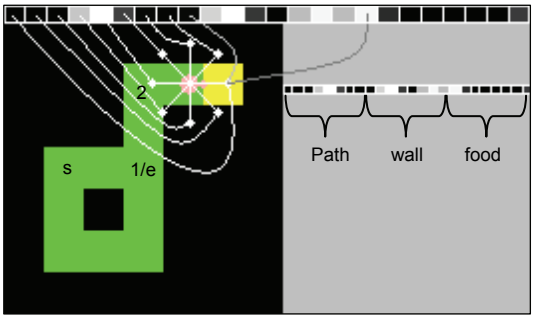

Figure 6.1. The experimental grid world. The agent is the "circle with nose" in the top right of the maze, In this figure the agent is looking to the right. The 8 white dots denote the points perceived by the agent. These points are connected to the elements of state *s* (neural input to the MLPs used by the agent) as depicted. This is repeated for all possible features, in our case: path (gray), wall (black), and food (light gray), in that order (as depicted in the smaller representation of the neural network). The "e" denotes the cell in which social reward can be administered through smiling or expression of fear, the "1" and "2" denote key locations at which the agent has to learn to differentiate its behavior, i.e., either turn left ("1") or right ("2"). The agent starts at "s". The task enforces a non-reactive best solution (by which we mean that there is no direct mapping from reward to action that enables the agent to find the shortest path to the food). If the agent would learn that turning right is good, it would keep walking in circles. If the agent learns that turning left is good, it would not get to the food

The agent learns to find the path to the food, and optimizes this path. At every step the agent takes, the agent updates its model of the expected benefit of a certain action as follows. It learns to predict the value of actions in a certain perceived state *s*, using an adapted form of *Q*-learning. The value function, $Q_a(s)$, is approximated using a multilayer perceptron (MLP), with 3x8=24 input, 24 hidden, and one output neuron(s), with *s* being the real-valued input to the MLP, *a* the action to which the network belongs, and the output neuron converging to $Q_a(s)$. As a result, every action of the agent (5 in total: forward, left, right, left and forward, right and forward) has its own network (see also Gadanho, 1999). The output of the action networks are used as action values in a standard Boltzmann action-selection function (Sutton & Barto, 1998). An action network is trained on the *Q*-value—i.e., $Q_a(s) \leftarrow Q_a(s)+\alpha(r+\gamma Q(s')-Q_a(s))$—where *r* is the reward resulting from action *a* in state *s*, *s'* is the resulting next state, $Q(s')$ the value of state *s'*, α is the learning rate and γ the discount factor (Sutton & Barto, 1998).

The learning rate equals 1 in our experiments (because the learning rate of the MLP is used to control speed of learning, not α), and the discount factor equals 0.99. To cope with a continuous grid world, we adapted standard *Q*-learning in the following way:

First, the value $Q_a(s)$ used to train the MLP network for action *a* is topped such that $min(r, Q_a(s'))<=Q_a(s)<=max(r, Q(s'))$. As a result, individual $Q_a(s)$ values can never be larger or smaller than any of the rewards encountered in the world. This enables a discount factor close to or equal to 1, needed to efficiently propagate back the food's reward through a long sequence of steps. In continuous, cyclic, worlds, training the MLP on normal *Q*-values using a discount factor close to 1 can result in several problems not further discussed here.

Second, per step of the agent, we train the action-state networks not only on $Q_a(s) \leftarrow Q_a(s) + \alpha(r+\gamma Q(s') - Q_a(s))$ but also on $Q_a(s') \leftarrow Q_a(s')$. The latter seems unnecessary but is quite important. RL assumes that values are propagated *back*, but MLPs generalize while trained. As a result, training an MLP on $Q_a(s)$ also influences its value prediction for *s'* in the same direction, just because the inputs are very close. In effect, part of the value is actually propagated *forward*; credit is partly assigned to what comes next. This violates the RL assumption just mentioned. Note that the value $Q(s')$ is predicted using another MLP, called the value network, that is trained in the same way as the action networks using the topped-off value and forward propagation compensation.

Third, for the agent to better discriminate between situations that are perceptually similar, such as position "1" and "2" in Figure 1, for each action-network the agent also uses a second network trained on the value of *not* taking the action. This network is trained when other actions are taken but not when the action to which the "negation" network belongs is taken. In effect, the agent has two MLPs per action. This enables the agent to better learn that, e.g., "right" is good in situation "2" but *not* in situation "1". Without this "negation" network, the agent learns much less efficient (results not shown). To summarize, our agent has 5 actions, it has 11 MLPs in total: one to train $Q(s)$, 5 to train $Q_a(s)$ and 5 to train $Q_{-a}(s)$. All networks use forward propagation compensation and a topped-off value to train upon. The MLP predictions for $Q_a(s)$ and $Q_{-a}(s)$ are simply added, and the result is used for action selection.

To study the effect of communicated affect as social reward, we created the following setup. First an agent is trained without social reward. The agent repeatedly tries to find the food for 200 trials, i.e., one *run*. The agent continuously learns and acts during these trials. To facilitate learning, we use a common method to vary the MLP learning rate and the Boltzmann action

selection β derived from simulated annealing. The Boltzmann β equals to 3+(*trial*/200)*(6−3), effectively varying from 3 (exploration) in the first trial to 6 (exploitation) in the last. The MLP learning rate equals 0.1−(*trial*/200)*(0.1−0.001) effectively varying from 0.1 in the first trial to 0.001 in the last. We repeated the experiment 200 times, resulting in 200 runs. Average learning curves are plotted for these 200 runs using a linear smoothing factor equal to 6 (Figure 6.2).

Second, a new agent is trained *with* social reinforcement, i.e., a human observer looking at the agent with his/her face analyzed by the agent, translating a smile to a social reward and a fearful expression to a social punishment. Again, average learning curves are plotted using a linear smoothing factor equal to 6, but now based on the average per trial over 15 runs (Figure 6.2). We experimented with three different social settings: (a) a moderate social reinforcement, r_{human}, from trial 20 to 30, where the social reinforcement is either −0.5 or 0.5 (happy vs. fearful, respectively); (b) a strong social reinforcement, r_{human}, from trial 20 to 25 where social reinforcement is either −2 or 2, i.e., more extreme social reinforcement but for a shorter period; (c) a social reinforcement, r_{human}, from trial 29 to 45 where social reinforcement is either −2 or 2 while (in addition to settings *a* and *b*) the agent trains an additional MLP to predict the direct social reinforcement, r_{human}, based on the current state *s*. The MLP is trained to learn $R_{social}(s)$ as given by the human reinforcement r_{human}. After trial 45, the direct social reinforcement from the observer, r_{human}, is replaced by the learned social reinforcement $R_{social}(s)$. So, during the critical period (the trial intervals mentioned) of social setting *a*, *b* and *c*, the total reinforcement is a composite reward equal to $R(s)+r_{human}$. Only in setting *c*, and only after the critical period until the end of the run, the composite reward equals $R(s)+R_{social}(s)$. In all other periods, the reinforcement is as usual, i.e., $R(s)$. As a result, in setting *c* the agent can continue using an additional social reinforcement signal that has been learned based on what its human tutor thinks about certain situations.

The process of giving affective feedback to a Reinforcement Learning agent appeared to be quite a long, intensive and attention absorbing experience. As a result, it was physically impossible to observe the agent during all runs and all trials in the entire grid world (after 2 hours of smiling to a computer screen one is exhausted *and* has burning eyes and painful facial muscles). To be able to test our hypothesis, we restricted social input to the cell indicated by 'e' (Figure 6.1). Only when the agent moves around in this cell (and is in a social input trial as defined by the social settings described above), the simulation speed of the experiment is set to one action per second enabling human affective feedback.

6.6 Results

The results clearly show that learning is facilitated by social reward. In all three social settings (Figure 6.2a, b and c) the agent needs fewer steps to find the food during the trials in which the observer provides assistance to the agent by expressing positive or negative affect. Interestingly, at the moment the observer stops giving social rewards, the agent gradually looses the learning benefit it had accumulated. This is independent of the size of the social reward (both social learning curves in Figure 6.2a and b show dips that eventually return to the non-social learning curve). This can be easily explained. The social reward was not given long enough for the agent to internalize the path to the food (i.e., propagate back the food's reward to the beginning of the path). As soon as the observer stops giving social rewards, the agent starts to forget these rewards, i.e., the MLPs are again trained to predict values as they are without social input. So, either the observer should continue to give social rewards until the agent has internalized the solution, or the agent needs to be able to build a representation of the social reward function and use it when actual social reward is not available. We have experimented with the second (social setting *c*): we enabled the agent to learn the social reward function. Now the agent uses actual social reward at the emotional input spot ('e', Figure 6.1) during the critical period, and uses its social reward prediction when social input stops. This is the third social setup. Results clearly show that the agent is now able to keep the benefit it had accumulated from using social rewards (Figure 6.2c). These results show that a combination of using social reward and learning a social reward function facilitates robot learning, by enabling the robot to quicker learn the optimal solution to the food due to the direct social reward as well as keep that solution by using its learned social reward function when social reward stops.

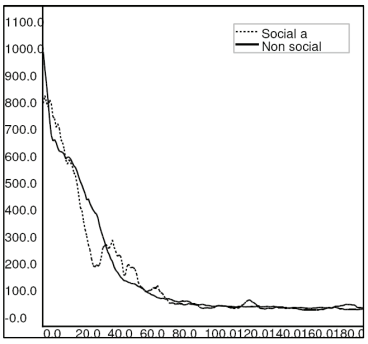

Figure 6.2a. Results of the learning experiment where the social setting *a* is compared with the non-social setting. In social setting *a*, social input is given between trial 20 and 30, where the social reward is either −0.5 or 0.5 (happy vs. fearful, respectively). On the *x*-axis the number of times the food is found is shown (trials); on the *y*-axis the average number of steps needed to find the food is shown.

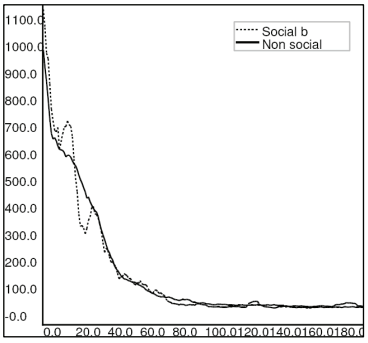

Figure 6.2b. Results of the learning experiment where the social setting *b* is compared with the non-social setting. In setting *b*, the social input is given between trial 20 and 25 where social reward is either −2 or 2, i.e., more extreme social rewards but for a shorter period. Axes are as in the previous figure.

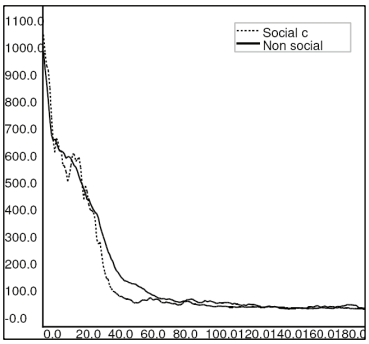

Figure 6.2c. Results of the learning experiment where the social setting *c* is compared with the non-social setting. In setting *c*, social input is given between trial 29 to 45, where social reward is either −2 or 2. The agent trains an additional MLP to predict the social reward. Axes are as in the previous figure.

6.7 Conclusion, Discussion and Further Work

Our results show that affective interaction in human-in-the-loop learning can provide significant benefit to the efficiency of a Reinforcement Learning robot in a continuous grid world. We believe our results are particularly important to human-robot interaction for the following reasons. First, advanced robots such as robot companions, robot workers, etc., will need to be able to adapt their behavior according to human feedback. For humans it is important to be able to give such feedback in a natural way, e.g., using emotional expression. Second, humans will not want to give feedback all the time, it is therefore important to be able to define critical learning periods as well as have an efficient social reward system. We have shown the feasibility of both. Social input during the critical learning periods was enough to show a learning benefit, and the relatively easy step of adding an MLP to learn the social reward function enabled the robot to use the social reward when the observer is away.

We have specifically used an experimental setup that is compatible with a real-world robot: we have used continuous inputs and MLP-based training of which it is known that it can cope with noise and generalize over training examples. We believe our results can be generalized to real-world robotics. However, this most certainly needs to be experimented with.

Many interesting computational approaches exist that study emotion in the context of robots and agents, of which we mention one explicitly here as it is particularly related to our work: the adaptive, social chatter bot *Cobot* (Isbell et al., 2001). Cobot learns the information preferences of its chat partners, by analyzing the chat messages for explicit and implicit reward signals. These signals are then used to adapt its model of providing information to that chat partner. So, Cobot effectively uses social feedback as reward, as does our simulated robot. However, there are several important differences. Cobot does not address the issue of a human observer parenting the robot using affective communication. Instead, it learns based on reinforcement extracted from words used by the user during the chat sessions in which Cobot is participating. Also, Cobot is not a real-time behaving robot, but a chat robot. As a consequence, time constraints related to the exact moment of administering reward or punishment are less important. Finally Cobot is restricted regarding its action-taking initiative, while our robot is continuously acting, with the observer reacting in real-time.

Future work includes a broader evaluation of the *EARL* framework including its ability to express emotions generated by an emotional model plugged into the RL agent. Further, it is interesting to experiment with controlling meta parameters (such as exploration/exploitation and learning rate) based on the agent's internal emotional state or social rewards, as has been done in the discrete grid-world case in Chapter 3 and 4. Currently we use simulated annealing-like mechanisms to control these parameters.

Further, the agent could try to learn what an emotional expression predicts. In this case, the agent would use the emotional expression of the human in a more pure form (e.g., as a real-valued vector of facial feature intensities as part of its perceived state *s*). This might enable the agent to learn what the emotional expression means for itself instead of simply using it as reward.

Finally, a somewhat futuristic possibility is actually quite close: affective Robot-Robot interaction. Using our setting, it is quite easy to train one robot in a certain environment (parent), make it observe an untrained robot in that same environment (child), and enable it to express its emotion as generated by its emotion model using its robot head, an expression recognized and translated into social rewards by the child robot. Apart from the fact that it is somewhat dubious if such a setup is actually useful (why not send the social reward as a value through a wireless connection to the child), it would enable robots to use the same communication protocol as humans.

Regarding the "usefulness" argument just put forward, it seems to apply to our experiment as well. Why didn't we just simulate affective feedback by

pushing a button for positive reward and pushing another for negative reward (or even worse, by simulating a button press)? From the point of view of the robot this is entirely true, however, from the point of view of the human—and therefore the point of view of the human-robot interaction—not at all. Humans naturally communicate social signals using their face, not by pushing buttons. The process of expressing an emotion is quite different from the process of pushing a button, even if it was only for the fact that it takes more time and effort to initiate the expression and that the perception of an expression is the perception of a process and not of a discrete event (like a button press). In a real-world scenario with a mobile robot in front of you it would be quite awkward to have to push buttons instead of just smile when you are happy about its behavior. Further it would be quite useful if the robot could recognize you being happy or sad, and gradually learn to adapt its behavior even when you did not intentionally give it a reward or punishment. Abstracting away from the actual affective interaction patterns between the human and the robot in our experiment would have rendered the experiment almost completely trivial. Nobody would be surprised to see that the robot learns better if an intermediate reward is given halfway its route towards food. Our aim was to investigate if affective communication can enhance learning in a Reinforcement Learning setting. Taking out the affective part would have been quite strange indeed.