Fear and Hope Emerge from Anticipation in Model-Based Reinforcement Learning

Thomas Moerland, Joost Broekens, and **Catholijn Jonker**
Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands
{T.M.Moerland,D.J.Broekens,C.M.Jonker}@tudelft.nl

Abstract

Social agents and robots will require both learning and emotional capabilities to successfully enter society. This paper connects both challenges, by studying models of emotion generation in sequential decision-making agents. Previous work in this field has focussed on model-free reinforcement learning (RL). However, important emotions like hope and fear need anticipation, which requires a model and forward simulation. Taking inspiration from the psychological Belief-Desire Theory of Emotions (BDTE), our work specifies models of hope and fear based on best and worst forward traces. To efficiently estimate these traces, we integrate a well-known Monte Carlo Tree Search procedure (UCT) into a model based RL architecture. Test results in three known RL domains illustrate emotion dynamics, dependencies on policy and environmental stochasticity, and plausibility in individual Pacman game settings. Our models enable agents to naturally elicit hope and fear during learning, and moreover, explain what anticipated event caused this.

1 Introduction

Robots and virtual agents will enter domestic environments in the forthcoming years. The unpredictability of real-world situations requires these agents to rely on learning, partly from interaction with non-expert humans. Moreover, acceptance by users requires them to have emotional and social capabilities as well. We belief emotions and learning/decision-making are best implemented together, as they provide mutual benefit (or even mutual necessity). On the one hand, connecting emotions to the decision-making process makes them relevant to the agent's goals and functionality. On the other hand, emotions can help the behavioural process in several ways, for example by influencing the agent itself (e.g. steering action selection [Broekens *et al.*, 2007]), communicating the current agent state to a social companion (i.e. transparency [Thomaz and Breazeal, 2008]) and by creating empathy and enhancing user investment. Note that we do not claim emotions are the only aspect of social communication, but the generality of these signals makes them a valuable research target for social learning settings.

To this end we will first need plausible models of emotion generation in sequential decision-making agents, which are studied in this paper. For the learning aspect we adopt computational reinforcement learning (RL), as it is a well-established approach to sequential decision making problems. Previous work on emotions in RL has focussed on model-free learning. However, important emotions like hope and fear are anticipatory, i.e. they require forward simulation. Moreover, while expressing anticipatory emotions may increase agent transparency, they will only make sense if the robot can explain which anticipated event caused them. As such, a social and emotional robot will need model-based learning and forward simulation.

The current work introduces the first anticipatory models of hope and fear in a RL agent. We ground our models in the psychological Belief-Desire Theory of Emotion (BDTE) [Reisenzein, 2009]. In particular, we show how hope and fear can be efficiently estimated from the best and worst forward traces. Subsequently, our results show the plausibility of these signals in three known RL domains.

The remainder of this paper is organized as follows. In section 2 we introduce model-based reinforcement learning, extending a known architecture (Dyna-2) with efficient forward planning (UCT). Section 3 introduces our emotional models based on the work by Reisenzein. In section 4 we investigate emotion dynamics in three known RL tasks: the Taxi domain (illustrating hope), Cliff Walking (illustrating fear) and the more complex and partially observable Pacman game. Sections 5 and 6 provide a discussion and conclusion of our work.

2 Model-based Reinforcement learning

Reinforcement learning (RL) has shown important success in robotic applications [Kober *et al.*, 2013]. However, its applications to social robotics remain relatively limited. Most reinforcement learning approaches have focused on model-free learning. Such methods circumvent the problem of learning the state transition function, as well as having to plan in larger state-spaces. However, model-based methods (reviewed in [Hester and Stone, 2012] and [Nguyen-Tuong and Peters, 2011]) also have specific advantages, like increased sample efficiency (by incorporating planning updates) and targeted exploration. In section 3 we will show how they provide emotion estimates as well.

A Markov Decision Process (MDP) is defined by the tuple: $\{S, A, R, P, \gamma\}$, where S denotes a set of states, A a set of actions, $R : S \times A \times S \to \mathbb{R}$ the reward function, $P : S \times A \times S \to [0, 1]$ the transitions function and $0 \leq \gamma \leq 1$ a discount parameter. The goal of the agent is to find a policy $\pi : S \times A \to [0, 1]$ maximizing the expected return in the environment:

$$Q^{\pi}(s,a) = \mathrm{E}_{\pi}\Big[\sum_{t=0}^{\infty} \gamma^t \; r(s_t, a_t, s_{t+1}) | s_0 = s, a_0 = a\Big] \quad (1)$$

Apart from estimating this action-value function, model-based RL methods also learn the environment dynamics (P) and reward function (R). An early model-based RL architecture was Dyna [Sutton, 1991] which, in between true sampling steps, randomly updates $Q(s, a)$ pairs. Shortly afterwards, this approach was made more efficient by prioritized sweeping [Moore and Atkeson, 1993], which tracks the $Q(s, a)$ tuples which are most likely to change, and focusses its computational budget there. More recently, Dyna with prioritized sweeping was combined with function approximation as well [Sutton *et al.*, 2012].

However, all these model-based approaches only update $Q(s, a)$ pairs in (recent) history in between true sample experience. Another extension of Dyna, called Dyna-2, also specifically includes *forward* sampling from the current node [Silver *et al.*, 2008]. In particular, Dyna-2 maintains two action value functions: $Q(s, a)$, estimated from true sample experience, and $\bar{Q}(s, a)$, estimated from forward sampling.

As the Dyna-2 architecture explicitly incorporates forward sampling (i.e. anticipation), we will adopt it in this work. However, we replace the forward *TD-sampling* of Dyna-2 with the successful planning procedure Upper Confidence Bounds for Trees (UCT) [Kocsis and Szepesvári, 2006]. UCT adaptively samples forward traces, which we later show to provide emotion signals as well. The remainder of this section covers our Dyna-2 extension by formally introducing value function estimation (2.1) and UCT planning (2.2).

2.1 Value function estimation

In each sample step, we observe the tuple $\{s, a, r, s'\}$. For this work, we implement a simple tabular learning method with $P(s'|s, a)$ and $R(s, a, s')$ estimated as normalized transition counts and average observed rewards, respectively.

The Q estimate of the particular (s, a) couple is then updated according to the (optimal) Bellman equation:

$$Q(s,a) = \sum_{s'} P(s'|s,a)[R(s,a,s') + \gamma \max_{a'} Q(s',a')] \quad (2)$$

where we act greedy in the update (off-policy). To further backup a possible change we implement prioritized sweeping [Moore and Atkeson, 1993]. Each state action pair (denoted by $\dot{s}, \dot{a}$) with a positive transition probability to the current node, is added to the queue with priority:

$$\rho(\dot{s},\dot{a}) = P(s|\dot{s},\dot{a}) \; \gamma[\max_{a} Q(s,a) - \max_{z} Q^{old}(s,z)] \quad (3)$$

where Q^{old} refers to the old Q estimate before the update in equation 1. We then update the highest priority pair, and repeat this process for a certain computational budget. Thereby, we directly spread out the contribution of a new observation over the state-space. This is important for our anticipatory emotion models (section 3), since it prevents anticipation artefacts due to postponed one-step back-ups. Finally, the behavioural policy directly follows from the Q estimates: $\pi(s, a)$ = ϵ-greedy(Q).

2.2 Planning

We now introduce the estimation of $\bar{Q}(s, a)$ through forward planning from the current node. We could plan by exploring all possible paths from the current node up to a specific horizon d. However, such Full Finite Horizon Planning (FFHP) approach grows exponentially in the search depth and also suffers from low probability transitions (i.e. high branching). Therefore, there has been increasing interest in Monte Carlo Tree Search (MCTS) methods, which built more effective tree search through roll-outs. A successful algorithm in this class is probably Upper Confidence Bounds for Trees (UCT) [Kocsis and Szepesvári, 2006]. UCT builds an adaptive tree based on previous estimates and the upper bound of the confidence interval for each available action. It thereby adaptively focusses on more promising or less explored actions. The algorithm has shown great success in the game Go [Wang and Gelly, 2007].

UCT samples N trajectories of depth $d_{\max}$. It maintains two counts: $B(s, d)$, i.e. the number of visits to state s at depth d, and $B(s, a, d)$, which is the state-action equivalent count. For each next sampling step in the known part of the tree, it chooses its action based on:

$$a_d = \arg\max_{a} \quad \bar{Q}_d(s,a) + c\sqrt{\frac{\ln(B(s,d))}{B(s,a,d)}} \quad (4)$$

where $\bar{Q}_d(s, a)$ denotes the estimate for node (s, a) at depth d obtained from previous traces, and c is a constant. When one of the available actions is unvisited (i.e. $B(s, a, d) = 0$), we take the second term in (4) as ∞, which implies a random selection among unseen actions. We then extend the tree with this new action and a sampled next state $s' \sim P(s'|s, a)$, and perform a targeted roll-out from there up to depth $d_{\max}$ following policy π. We then back-up the nodes in the tree, and repeat this process N times. Thereby, each trace expands the tree by one action-new-state combination. By adaptively selecting the best returning paths, we focus new roll-outs to more promising area's.

The UCT algorithm is intended to provide efficient estimates of $Q(s, a)$ at the root node. These estimates, denoted by $\bar{Q}(s, a)$, can be used for action selection as well (e.g. $\bar{\pi}(s, a)$ = ϵ-greedy($\bar{Q}$)). We will not consider incorporating $\bar{Q}$ in action selection in this work. However, we did intentionally introduce these planning concepts separately, as the proposed framework is essentially a combination of well-established procedures (Algorithm 1). In the next section we will show how this procedure will give us emotion estimates at little additional computational expense.

Algorithm 1 Model-based reinforcement learning with emotion simulation.

```
Initialize $R, P, Q, \varrho$
while s not terminal do
    Emotion, $\bar{Q}(s, a) \leftarrow$ UCT($s$,$d$,$N$) // section 2.2
    $a \leftarrow \pi(s, a)$ // (could use $\bar{\pi}(s, a)$)
    Execute $a$, observe $r$, $s'$
    $R, P \leftarrow$ UpdateModel($s$,$a$,$r$,$s'$) // section 2.1
    $Q, \varrho \leftarrow$ PrioritizedSweeping($Q$, $\varrho$) // section 2.1
    $s \leftarrow s'$
end while
```

3 Emotion

Emotion has been extensively linked to decision-making in both psychology (reviewed in [Baumeister *et al.*, 2007]) and neuroscience (reviewed in [Rolls and Grabenhorst, 2008]). Computational implementations of emotion generation in RL have used homeostasis [Gadanho and Hallam, 2001], appraisal dimensions [Sequeira *et al.*, 2011] and reward or value functions [Broekens *et al.*, 2007; Salichs and Malfaz, 2012]. However, none of the mentioned emotion models in a RL or MDP context consider anticipation. Decision making research has shown anticipation of future emotions is an important aspect in human choice [Mellers *et al.*, 1999]. Other computational emotion models, like those based on cognitive appraisal theory (reviewed in [Gratch and Marsella, 2014]), also do consider anticipation, but do no consider learning.

Recently, [Broekens *et al.*, 2015] did introduce a model of hope and fear in a learning context, but the authors derive these anticipatory emotions from the current state value. However, the value is an average of the return over all future traces, i.e. it averages potential good and bad outcomes. However, emotions like hope and fear rather focus on specific future events, which can only be obtained by 'splitting up' the value of the current state in positive and negative components.

We base this observation in the psychological belief-desire theory of emotion (BDTE) [Reisenzein, 2009], which conceptualizes the origin of seven emotions (joy, distress, hope, fear, surprise, disappointment and relief) based on two underlying dimensions. The theory handles reasoning about a particular 'state of affairs' (i.e. s). BDTE then defines the *belief* about the state, $b(s) \in [0, 1]$, as the (subjective) probability that the state will come true, and the *desire* of the state, $d(s) \in \mathbb{R}$, as the desirability of the s. The seven emotions relate to a partitioning of the belief-desire space, while the emotion intensity is defined by the product of belief and desire, i.e. $I(s) = b(s) \times d(s)$. This work will interpret the partitioning for joy, distress, hope and fear.

3.1 Joy and distress

Based on BDTE, joy occurs when $d(s) > 0$ and $b(s) = 1$, i.e. when the state is both desirable and absolutely certain. In a MDP context, we can only be certain about our *current* experienced transition. The desirability of this transition is best captured by the temporal difference (TD). Note that we should not only consider the experienced reward, as this specification is sensitive to a translation of the reward function, while the TD is robust against such shifts. We therefore define the joy J upon arriving in state s' as:

$$\begin{aligned} J(s, a, s') &= b(s') \times d(s, a, s') \\ &= \Big[r(s, a, s') + \gamma \max_z Q(s', z) - Q(s, a) \Big]^+ \end{aligned} \tag{5}$$

where $b(s')$=1 and $+$ denotes the positive part. Distress $D(s, a, s')$ is analogously defined as the negative part of the last expression of (5).

The relation between happiness (or dopamine expression) and the TD is actually well researched in neuroscience (e.g. reviewed in [Rolls and Grabenhorst, 2008]). Previous computational work has also implemented this idea [Broekens *et al.*, 2015]. We mainly show this specification directly follows from BDTE as well. Furthermore, note that with converged action value estimates, we can still be happy or unhappy about an event if the transition function is stochastic. For example, if a particular action has both transitions to a positive and negative event, this will balance in the estimate of $Q(s, a)$ (see equation 1), but the agent will still experience joy or distress depending on the actually *experienced* transition.

3.2 Hope and fear

Our main extension of previous work is including anticipatory hope and fear through forward planning. According to BDTE, hope and fear originate when $b(s) < 1$ (in combination with $d(s) > 0$ or $d(s) < 0$, respectively). Therefore, hope and fear emerge when we are still uncertain whether an event will happen. We propose this uncertainty refers to anticipation, as we can only be uncertain about future events. Since the interpretation of desirability remains similar to the previous paragraph, hope and fear become *anticipated temporal differences* related to a particular forward trace.

We will define the experienced hope and fear in the current state s as the future state with the best and worst product of likelihood and temporal difference. We denote a trajectory of depth d from the current node by $g_d = \{s_0 a_0 s_1 a_1 ... s_{d-1} a_{d-1} s_d\}$, and for readability we write $V(s) = \max_a Q(s, a)$. The hope in node s_0 is given by:

$$\begin{aligned} H(s_0) &= \max_{s'} \Bigg[b(s'|s_0) \times d(s'|s_0) \Bigg] \\ &= \max_{s'} \sum_{d, g_d | s_d = s'} \Bigg[\prod_{t=0}^{(d-1)} \pi(s_t, a_t) P(s_{t+1}|s_t, a_t) \times \\ &\quad \Big(\big[\sum_{t=0}^{(d-1)} \gamma^t r(s_t, a_t, s_{t+1}) \big] + \gamma^d V(s_d) - V(s_0) \Big) \Bigg]^+ \end{aligned} \tag{6}$$

Note that we sum over all traces towards node s', as there might be multiple paths towards the feared or hopeful event. Furthermore, note that the uncertainty of each trace depends both on our own choices, π, and environmental stochasticity,

$P(s'|s,a)$. Obviously, d will in practice have to be bounded by a search horizon d_{max}.

The specification of fear is given by changing the maximization in a minimization:

$$F(s_0) = \min_{s'} \Big[b(s'|s_0) \times d(s'|s_0) \Big]$$
$$= \min_{s'} \sum_{d, g_d | s_d = s'} \Bigg[\prod_{t=0}^{(d-1)} \pi(s_t, a_t) P(s_{t+1}|s_t, a_t) \times$$
$$\Big(\big[\sum_{t=0}^{(d-1)} \gamma^t r(s_t, a_t, s_{t+1}) \big] + \gamma^d V(s_d) - V(s_0) \Big) \Bigg]^{-} \tag{7}$$

We will use UCT to find the hope and fear signals, by adaptively sampling forward traces. As we want to sample close to the behavioural policy, we replace $\bar{Q}(s,a)$ with $Q(s,a)$ in equation 4. Then, for each UCT beam, we extend the search tree by a new state-action combination, for which we evaluate the emotion for *all* s'. In conclusion, this forward procedure estimates the hope and fear in a current state, while enabling the agent to explain what it is hopeful or afraid for (i.e. state s').

4 Experiments

We test the emotion models in three scenario's: the Taxi domain (4.1) for hope, joy and distress, the Cliff Walking scenario (4.2) for fear, and finally Pacman (4.3) for plausibility of signals in a more complex and partially observable task.

4.1 Taxi Domain

In the fully observable Taxi domain (figure 1, introduced in [Dietterich, 1998]) the agent picks-up and drops-off a passenger. To simplify interpretation, we fix pick-up and drop-off location at R and B (respectively), while the taxi starts at a random location. The available actions in each state are: {N, S, E, W, Pick-up, Drop-off}. The episode ends with a correct delivery of the passenger ($r = +1$). However, this drop-off fails in 30% of cases ($r = -0.1$). All other transitions are deterministic and have a small penalty ($r = -0.01$).

Figure 2 shows the reward, joy/distress and hope signal for the first 700 iterations of the agent. The top graph shows the reward signal, with the agent solving the task for the first time around iteration 400. Before that time, there are some small joy and distress signals due to exploratory steps. Obviously, there is no hope signal yet, as we did not experience anything better than our initialization ($Q(s,a) = 0$). However, when the agent first solves the task just after iteration 400, we still do not observe any hope signals. This happens because to the agent, the environment is still fully deterministic, and therefore all good outcomes are expected. This illustrates a situation in which a task seems mastered, which makes hope irrelevant. However, around iteration 500 the agent encounters the negative event of a failing drop-off. It now learns the last action is actually stochastic, and therefore starts experiencing hope upon approaching the target (as observed in the cyclic hope signal after iteration 500 in figure 2).

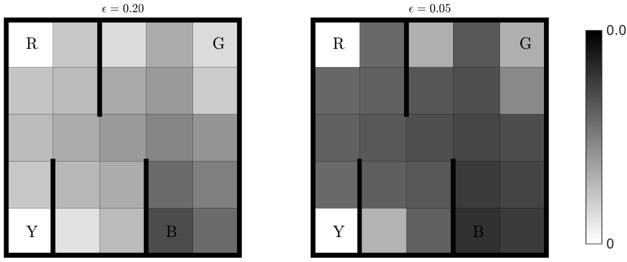

Figure 1: Taxi domain. Hope experienced by the agent per location when the passenger is in the taxi for ϵ-greedy policy with ϵ=0.20 (left) or ϵ=0.05 (right). The ϵ=0.05 agent has less exploration, which makes it more hopefull about reaching the target. Results for UCT(N=300, $d_{\max}$=7).

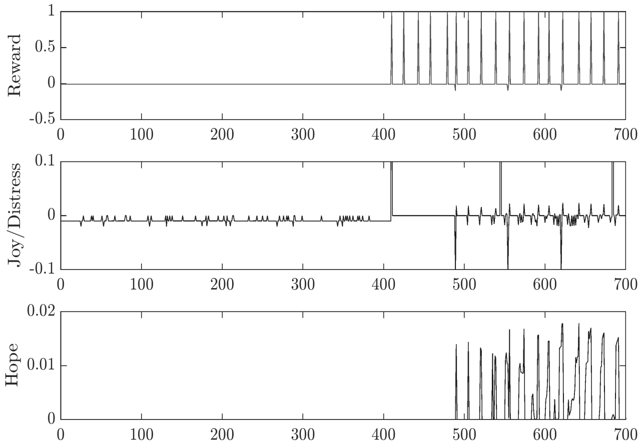

Figure 2: Reward, joy/distress, hope and fear for the first 700 iterations in the Taxi domain with UCT(N=300, $d_{\max}$=7).

Figure 1 shows the hope signal per grid location after learning a stable world model. Note that the passenger is already in the Taxi, so the agent needs to deliver it at location B. Results are shown for two behavioural policies. The agent smoothly estimates the hope for most grid locations. The largest distance to the goal is 8 steps (7 moves plus one drop-off), so the top-left and bottom-left location are blind for the target TD (i.e. no hope). Note that a full horizon search with TD evaluations at each depth involves at least $\sum_{i=1}^{d_{max}} k^i$ traces, for k available actions per state (ignoring any stochasticity in the transition function). In this small example, this would already involve $\sum_{i=1}^{7} 6^i > 3 \cdot 10^5$ forward beams. In that perspective, the agent seems to capture the hope signal well with only 300 traces. However, the hope TD is expected to hide in the direction of the behavioural policy, making it relatively easy to find. In the next section we will address the harder problem of finding the largest fear signal.

4.2 Cliff Walking

In the Cliff Walking scenario (figure 3, adopted from p.149 of [Sutton and Barto, 1998]) the environment has a specific negative event. The agent should walk from Start to Goal, but

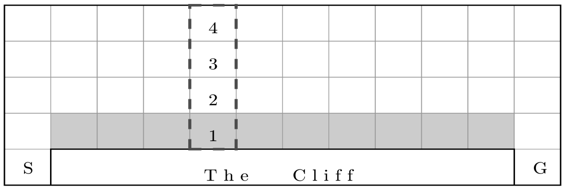

Figure 3: Cliff Walking scenario. Slippery path marked in grey (see text). The red box indicates the distance 4,3,2 and 1 to the Cliff depicted on the horizontal axis of figure 4.

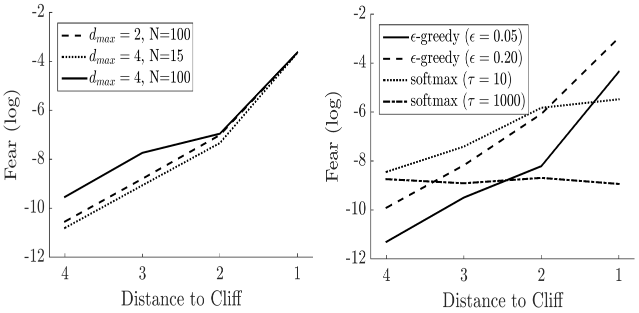

Figure 4: Left: Effect of search depth d_{max} and number of traces N in UCT. Right: Effect of different behavioural policies on the fear signal. Distance to cliff (horizontal axis) is illustrated in figure 3. Results averaged over 100 UCT($N =$ 18,$d_{max} = 4$) runs on a converged model.

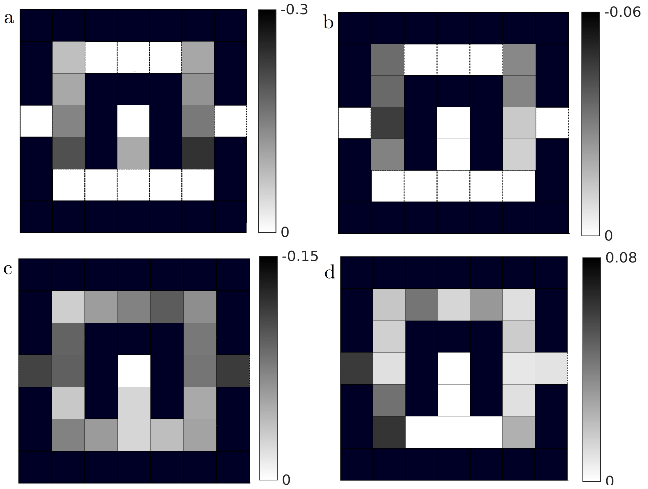

Figure 5: Pacman. a) Fear per location for a ghost *below* and ϵ-greedy(0.10) policy. b) Idem for softmax(τ=10) policy. c) Fear per location for *no* ghost and ϵ-greedy(0.10) policy. d) Hope per location for *no* ghost and softmax(τ=10) policy.

the shortest path moves along a Cliff. The agent can move in each of the cardinal directions. An episode ends when the agent reaches the Goal ($r = +0.02$) or falls into the Cliff ($r = -1$). All other transitions have a small penalty ($r = -0.001$). Along the Cliff, there is a 'slippery path' (marked in grey in figure 3). When stepping on these locations, the agent slips away to each cardinal direction with probability 0.01 (i.e. risking slipping into the Cliff). Due to the slippery path, the optimal policy leads along the middle row.

The fear signal poses a challenge, since we need to balance following the policy (for the probability of the event occurring), but occasionally need to explore some less like actions to find large temporal differences (e.g. moving on the slippery path). Therefore, we modify equation 4 from a strict maximization to an ϵ-greedy policy with ϵ=0.10. This shows slightly improved fear estimates (results not shown). Note that we modify the *search* policy (i.e. in the tree, equation 4), which is different from the behavioural policy that determines the $\pi(s, a)$ in equation 6 and 7.

The left plot of figure 4 shows the effect of the UCT depth (d_{max}) and number of traces (N) on fear estimation. We can see how the UCT(d_{max}=4,N=100) agent accurately detects the Cliff at all distances from it. However, both the UCT(d_{max}=2,N=100) and UCT(d_{max}=4,N=15) sometimes fail to detect the Cliff at distance 3 and 4, due to short-sightedness and lack of traces respectively.

The right plot of figure 4 shows the effect of different behavioural policies π, which is used to determine the action selection probabilities in the path in equation 6 and 7. Most noteworthy, we can see a different fear dynamic for ϵ-greedy versus softmax action selection. The ϵ-greedy agent has a much steeper increasing fear line, i.e. it is especially afraid just next to the cliff. This was to be expected, as ϵ-greedy has a uniform probability on all non-greedy actions, and it should therefore be very afraid to just walk into the cliff. On the contrary, the softmax agent does not consider stepping into the cliff at all. However, it does consider moving to the slippery path (which is not greedy, but still has reasonable Q estimates), which causes it to have higher fear at larger distances from the cliff. This illustrates the dependency of the signals on the behavioural policy.

4.3 Pacman

The previous scenario's investigates hope and fear dynamics and their dependency on environmental stochasticity (P), policy (π) and simulation parameters. Our final scenario will evaluate the plausibility of these signals in a more complex scenario: Pacman (figure 5, based on [Sequeira *et al.*, 2014]). Pacman (starting from center-top) should capture the 'power-pellet' (located in the center square). The ghost starts from the power-pellet location, and moves towards Pacman with $Pr = 0.8$ and random otherwise. An episode ends when the ghost captures Pacman ($r = -1$) or when Pacman reaches the power-pellet ($r = +1$). Pacman's state consists of location and whether a ghost exists in each corridor (4 directions), but not the distance to it. He can move $\{$up,down,left,right$\}$. We let Pacman interact with the environment for 100000 iterations (ϵ linear decreasing from 1 to 0.05 in the first 30000 iterations). Subsequently, we evaluate the plausibility of Pacman's emotions in several game scenario's (figure 5).

Figure 5a shows Pacman's fear when he observes a ghost below (only possible in vertical corridors), for ϵ-greedy policy. Pacman has learned to have most fear near the bottom of the corridor, as the ghost must be directly below it (remember that it cannot see the distance to the ghost). Figure 5b compares Pacman's fear for the same situation with a softmax behavioural policy. Pacman is now relatively less fearful in the bottom locations, as its policy strongly indicates moving up there (i.e. away from the ghost). However, higher up in the corridor the Q-values for up and down are more alike, since it could be beneficial to move down (and e.g. cross sides), but this obviously also causes fear. This is a nice illustration of 'splitting-up' the value function, i.e. it contains both good and bad traces. Also note how fear just below the power-pellet has fully disappeared (as the Q-values with softmax policy make it greedy to grab the power-pellet there). Figure 5c shows the fear experienced by Pacman when it does *not* observe a ghost in any direction. At nearly all locations Pacman fears a capture by the ghost, but this is especially prominent just before the corners (e.g. at the most left and right grid point). Pacman starts to feel safer when he gets closer to the power-pellet.

Finally, 5d shows the hope experienced by Pacman when he does not see a ghost. We would expect Pacman to be more hopeful towards the power-pellet location, but something odd happens here. When Pacman nearly reached the power-pellet location, it is not so hopeful at all, as it is almost fully expecting to obtain it. It turns out to be most hopeful just before the corners, hoping to step around it and still not see the ghost. This implies a jump in the value function, and happens on both sides of the top corridor, at the bottom of the left corridor, and also on the far left (note how Pacman has developed a left-wing tactic). At the bottom left location Pacman's hope is to advance another step horizontally without seeing a ghost, which makes it unlikely the ghost was hiding near the power-pellet. Altogether, these unexpected results do illustrate plausibility of the signals, as Pacman identified specific locations where things might change for better or worse.

5 Discussion

Compared to previous computational models that functionally ground emotions in a learning framework, we are the first to explicitly take anticipation into account. The closest related approach is probably the fearful robot Maggie [Salichs and Malfaz, 2012]. Maggie stores a fear per state as the worst Q-value it historically associated with it, remembering a location where something bad ever happened. Our implementation agrees with this view of fear as the worst event that might happen. However, the Q-value is an average over all outcomes (usually based on greedy back-ups). As we discussed before, the Q-value is therefore a bad predictor of individual malicious traces, nor does it take our own behavioural policy into account. Moreover, Maggie has no capability of anticipating the feared event (until one step before it).

Our models of emotion generation are themselves a contribution to the affective computing literature. We identify two main directions for future work, related to the different emotion functions mentioned in the introduction. First, our emotional signals should be evaluated in human-robot interaction (HRI) settings, both for their potential to communicate the decision-making process, and for their potential to enhance empathy and user investment. Second, the emotional signals can also modify the agent's learning process, for example by biasing action selection (i.e. creating agents that explicitly target and avoid more extreme events).

As this work used smaller scenario's and tabular learning, we cannot claim it directly generalizes to social agents with large state-spaces. We intend to introduce function approximation in future work, as our methodology remains theoretically just as applicable in large domains. Moreover, the generalization of function approximation might also make anticipation and forward planning much more realistic (i.e. with the agent anticipating events it has not seen itself, a feature that is impossible with tabular learning).

Another important challenge of model-based RL is quantification of model uncertainty. Our framework needs prioritized sweeping to spread out the temporal differences across the state-space (see section 2.1). For this work we naively implemented back-ups after limited experience, which poses the risk of converging to local minima in more complicated environments. A good generic framework to quantify model uncertainty is 'Knows What It Knows' (KWIK), which was already combined with UCT planning [Walsh *et al.*, 2010].

Finally, our work also illustrates another difference between 'laboratory' and 'social' RL. While laboratory settings allows engineered reward functions and any (naive) exploration strategy (e.g. ϵ-greedy), social scenario's make reasonable policies and proper reward scaling vital. For example, in Cliff Walking (4.2), the ϵ-greedy agent fears walking into the Cliff itself. Although this provided a nice illustration here, it is not very reasonable in real life. More subtle strategies, like softmax, require tedious numerical scaling, but provide more realistic social signals. The same argument holds for the reward function. For example, how much worse is 'falling in the cliff' compared to 'taking a step'? In our early implementations, the ratio between the two was too small (actually, even the current ratio of 1000 is debatable). This caused the agent at larger distance from the Cliff to fear 'making a step back' (small TD but quite likely) although it had detected the Cliff (large TD but less likely path). This happens because the influence of the probability multiplications in the path can become quite strong, but the real problem is that we should have judged falling in the cliff much worse. In conclusion, social scenario's will force RL research to start focussing more on subtle policies, and reward functions that comply with real-world consequences.

6 Conclusion

This paper introduced the first anticipatory models of hope and fear in a RL/MDP agent. We have shown how hope and fear can be efficiently estimated from adaptive forward traces, and illustrated the plausibility of these signals in several scenario's. As a first benefit of this approach, our emotions automatically emerge from the agent's functionality, without the need for any pre-wired (and ad-hoc) solutions. Moreover, by incorporating forward simulation, the agent can also explain the origin of its hope or fear. This might enable a social agent to naturally express its behavioural process, increase credibility and empathy and facilitate social interaction.

Acknowledgments

The authors want to thank the reviewers for their valuable comments, that helped improve the final version of this paper.

References

[Baumeister *et al.*, 2007] Roy F Baumeister, Kathleen D Vohs, C Nathan DeWall, and Liqing Zhang. How emotion shapes behavior: Feedback, anticipation, and reflection, rather than direct causation. *Personality and Social Psychology Review*, 11(2):167–203, 2007.

[Broekens *et al.*, 2007] Joost Broekens, Walter A Kosters, and Fons J Verbeek. Affect, anticipation, and adaptation: Affect-controlled selection of anticipatory simulation in artificial adaptive agents. *Adaptive Behavior*, 15(4):397–422, 2007.

[Broekens *et al.*, 2015] Joost Broekens, Elmer Jacobs, and Catholijn M. Jonker. A reinforcement learning model of joy, distress, hope and fear. *Connection Science*, pages 1–19, 2015.

[Dietterich, 1998] Thomas G Dietterich. The MAXQ Method for Hierarchical Reinforcement Learning. In *ICML*, pages 118–126. Citeseer, 1998.

[Gadanho and Hallam, 2001] Sandra Clara Gadanho and John Hallam. Robot learning driven by emotions. *Adaptive Behavior*, 9(1):42–64, 2001.

[Gratch and Marsella, 2014] J. Gratch and S. Marsella. Appraisal Models. In Rafael A Calvo, Sidney D'Mello, Jonathan Gratch, and Arvid Kappas, editors, *The Oxford Handbook of Affective Computing*, pages 54–67. Oxford University Press, 2014.

[Hester and Stone, 2012] Todd Hester and Peter Stone. Learning and using models. In *Reinforcement Learning*, pages 111–141. Springer, 2012.

[Kober *et al.*, 2013] Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, page 0278364913495721, 2013.

[Kocsis and Szepesvári, 2006] Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *Machine Learning: ECML 2006*, pages 282–293. Springer, 2006.

[Mellers *et al.*, 1999] Barbara Mellers, Alan Schwartz, and Ilana Ritov. Emotion-based choice. *Journal of Experimental Psychology: General*, 128(3):332, 1999.

[Moore and Atkeson, 1993] Andrew W Moore and Christopher G Atkeson. Prioritized sweeping: Reinforcement learning with less data and less time. *Machine Learning*, 13(1):103–130, 1993.

[Nguyen-Tuong and Peters, 2011] Duy Nguyen-Tuong and Jan Peters. Model learning for robot control: a survey. *Cognitive processing*, 12(4):319–340, 2011.

[Reisenzein, 2009] Rainer Reisenzein. Emotional experience in the computational belief–desire theory of emotion. *Emotion Review*, 1(3):214–222, 2009.

[Rolls and Grabenhorst, 2008] Edmund T Rolls and Fabian Grabenhorst. The orbitofrontal cortex and beyond: from affect to decision-making. *Progress in neurobiology*, 86(3):216–244, 2008.

[Salichs and Malfaz, 2012] Miguel Angel Salichs and María Malfaz. A new approach to modeling emotions and their use on a decision-making system for artificial agents. *Affective Computing, IEEE Transactions on*, 3(1):56–68, 2012.

[Sequeira *et al.*, 2011] Pedro Sequeira, Francisco S Melo, and Ana Paiva. Emotion-based intrinsic motivation for reinforcement learning agents. In *Affective computing and intelligent interaction*, pages 326–336. Springer, 2011.

[Sequeira *et al.*, 2014] Pedro Sequeira, Francisco S Melo, and Ana Paiva. Emergence of emotional appraisal signals in reinforcement learning agents. *Autonomous Agents and Multi-Agent Systems*, pages 1–32, 2014.

[Silver *et al.*, 2008] David Silver, Richard S Sutton, and Martin Müller. Sample-based learning and search with permanent and transient memories. In *Proceedings of the 25th international conference on Machine learning*, pages 968–975. ACM, 2008.

[Sutton and Barto, 1998] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 1998.

[Sutton *et al.*, 2012] Richard S Sutton, Csaba Szepesvári, Alborz Geramifard, and Michael P Bowling. Dyna-style planning with linear function approximation and prioritized sweeping. *arXiv preprint arXiv:1206.3285*, 2012.

[Sutton, 1991] Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM SIGART Bulletin*, 2(4):160–163, 1991.

[Thomaz and Breazeal, 2008] Andrea L Thomaz and Cynthia Breazeal. Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence*, 172(6):716–737, 2008.

[Walsh *et al.*, 2010] Thomas J Walsh, Sergiu Goschin, and Michael L Littman. Integrating Sample-Based Planning and Model-Based Reinforcement Learning. In *AAAI*, 2010.

[Wang and Gelly, 2007] Yizao Wang and Sylvain Gelly. Modifications of UCT and sequence-like simulations for Monte-Carlo Go. *CIG*, 7:175–182, 2007.