Affect and Learning: a computational analysis

Broekens, D.J.

1

Introduction

The research described in this computer science PhD thesis is positioned somewhere between computer science and psychology. It is about the influence of affect on learning. Affect is related to emotion; *affect* is about the *positiveness* and *negativeness* of a situation, thought, object, etc. We will define affect more precisely in Section 1.4, but for now this definition suffices. Affect can influence learning and behavior in many ways. For example, parents use affective communication to influence the behavior of their children (praise versus disapproval). Affect can also influence how individuals process information (e.g., positive affect favors creativity, negative affect favors critical thinking). The research described in this thesis uses computational modeling to study affective influence on learning. The goal has been twofold: first, understand more about potential mechanisms underlying relations between affect and learning as found in the psychological literature, and second, study if the concept of affect can be used in computer learning, most notably to control the learning process. Both aspects are considered of equal importance in this work. The topic is quite interdisciplinary and the individual chapters present the results of focused studies. However, in an attempt to clarify to a broader public what the research questions are and why these are of interest, the introduction is intentionally kept broad and is written so that it is understandable to readers with general knowledge of computer science and an interest in psychology. Readers that want to skip the introduction can read Section 1.5 for an overview of the thesis.

1.1 Informal Introduction to the Topic

This thesis is about affect and learning, a topic everyone is intuitively familiar with. We all know the effects of anger and sadness (two different negative affective states) or happiness and excitement (two different positive affective states) on our own functioning and decisions. Sometimes, we regret these decisions, while others worked out quite fine—better than expected—afterwards. In everyday life, we just accept that we have emotions and that our emotions influence our behavior. It is common sense knowledge that it is sometimes the head, sometimes the heart that decides our future and we rarely ask ourselves when and how affect exactly influences our decisions. Interestingly, it is quite difficult to reflect upon a decision, let's say the last decision you made, and discriminate between the "affect" part versus the "rational thought" part that influenced that decision. Instead, affect and "rational thought" seem to be intertwined in many cases, a notion put forward by Antonio Damasio in his seminal book *Descartes' Error* (Damasio, 1994). It is by now generally accepted that "rational thought" does not exist, at least not in the sense we thought (hoped?) it did. Nothing is decided purely based on a logical evaluation of pros and cons of which the pros and cons are again (recursively) a result of a logical evaluation of *their* pros and cons of which the pros and cons … etc. This kind of recursive and analytic thought process is very rare, chess-like game play being perhaps a partial exception, and it is by no means necessary for normal functioning in society; other animals don't need it either and are quite adaptive to their environment. What seems to be more the case is exemplified by the following "should I stay or should I go" scenario (also a nice song by *The Clash* showing human indecisiveness):

> I'm at work, writing the introduction of my thesis. Some chapters still have to be written, so quite some writing still has to be done. However, today actually is a local holiday called "Leids Ontzet" feasting the liberation of the city of Leiden (The Netherlands) ending the Spanish occupation of that city in the year 1574. The faculty is closed but I went in with my key to do some work. So, the decision is: should I stay the whole day and write as much as possible, or should I go home and do something else taking advantage of the fact that today is a local holiday. Now here's my "rational choice": I went to work this morning, because I am not originally from Leiden, so I do not really care about Leids Ontzet. My partner also went to work, because she works in The Hague (not in Leiden: thus no local holiday). I do care about playing video games in my spare time, and therefore I like having a day off. However, I do have a lot of work to do on my thesis, and I want my thesis to be finished in time. (Why? Because my supervisor wants me to? Because it is good for my future? Because it just feels like the right thing to do?). So, here am I, having to decide on two things: go home and play games (which I like), versus stay and write my thesis (which I like). It is a holiday, but my partner has to work. So, taking a day off *now* enables me to play games, but I won't be able to work on my thesis, and it takes away my option to take another day off when my partner does have a day off. What do I do? I work in the morning on my thesis, write a fair part of the introduction, and take the afternoon off and play games. I get to do two things I like, and keep the option of taking half a day off to do nice stuff with my girlfriend later, which I also like. So isn't this a win-win-win situation? It probably is, but the decision itself is not rational, it is emotional and social and there is no deep logical evaluation behind the value of the alternatives. The only thing that might be called rational is the process by which I generate the alternatives. However, the decision is made based on a "what feels best" criterion, and I just "weight" the values of the alternatives using social and emotional associations. One could even argue that I did not decide anything at all: none of the alternatives is excluded; instead I have chosen a mixture of things that feels good to me. Many decisions resemble this scenario, and I think we can agree that our life's course is a long sequence of such decisions, none of them being exclusively rational, none exclusively affective.

The question seems to be how and when affect influences decision making, thought, learning, and the many other cognitive phenomena known in cognitive psychology. For example, psychologists like Joseph Forgas, Alice Isen and Gerald Clore have studied the influence of emotion and affect on human decision making for quite some time (for references see Chapter 2). Although much debate is going on, as discussed for example in Chapter 3, decades of research indeed converged into a general consensus that affect *does* influence cognition in important ways. These ways include affect manipulating how we approach problems—e.g., do we look at the details of a problem, or approach it from the top—, affect influencing what we think about objects and people, and affect influencing creativity and open-mindedness.

Although much is known on the influence of affect on cognition, the mechanisms by which affect influences cognition are largely unknown. This is partly because it is very difficult to experimentally manipulate and subsequently measure affect, let alone affective influence on, for example, decision making and learning. This is exactly where the computer enters (fortunately, as this is a Computer Science PhD thesis, and some might at this point be wondering where the computer went). Computers enable scientists to develop computational models (programs) that can actually produce "new things", based on the assumptions of the theoretical model (e.g., a psychological theory describing the influence of affect on learning) underneath the computer model. These "new things" are, in a very real sense, predictions of the psychological theory: they result from the computational model that is a highly detailed version, an implementation, of the psychological theory. As such, computational models help psychological theory development. As computer models need to "run", they need to execute a sequence of commands and manipulate the results of these commands; computer models are particularly good at investigating mechanism, because they exist by the virtue of mechanism. Mechanism happens to be the thing that is notoriously difficult to investigate based on observation of behavior (whether that is body movement, data from brain scanners, facial expressions, or biochemical markers). We can thus conclude that computational modeling is a useful method to study potential mechanisms proposed by psychological and neurobiological theories, including theories about the influence of affect on learning.

This thesis presents research on the influence of affect on learning by means of computational modeling. As such, both affect and learning need to be computationally modeled. A successful model for task-learning is *reinforcement learning* (RL). It has been applied to many computer learning problems, such as computers that learn to play games, steer cars, and control robots (see Sutton &

Barto, 1998). The RL paradigm is quite analogous to instrumental conditioning. Instrumental conditioning is a paradigm by which animals (including humans) can learn new behaviors, by trying new actions (exploration) and receiving rewards and punishments (reinforcement) for these actions. Rewards and punishments can transfer to the actions the animal chose to do just before the action resulting in the reinforcement, and to actions before *that* action, and before *that* action, etc. As a result, the animal learns to execute a sequence of actions in order to get to a reward or avoid a punishment; the animal is said to exploit its knowledge after a period of exploration of its environment. Reinforcement Learning is a detailed computational model that describes how reinforcement can propagate back to earlier actions (this process of propagation is also known as *credit assignment*), as well as how the *values* of actions need to be adapted to reflect the received reinforcement (Section 1.3). Recently, neuroscientists have found evidence that parts of the human brain (and brains of other animals) seem to be involved in exactly this process of reward processing. The basal ganglia (an important dopamine system in the brain responsible for the initiation of action) are involved in the selection of actions, and neurons in the basal ganglia seem to encode the reinforcement signal, i.e., the change that needs to be made to the value of an action. Neurons in the prefrontal cortex (responsible for planning and executive, reflective processing) seem to encode the value (i.e., the effective credit a certain action is responsible for) of actions in a certain context. In studying learning, Reinforcement Learning seems to be a good candidate model; a point of view that is detailed in Section 1.3.

In Chapter 2 we introduce a measure for artificial affect that relates to a simulated animal's relative performance on a learning task (let's say, a simulated mouse in a maze searching for cheese). As such, artificial affect measures how well the simulated animal improves. Our animal learns by reward and punishment, thus, in our case, how "well" can be defined as the average reinforcement signal. Therefore the animal's performance can be defined as the difference between the long-term average reinforcement signal ("what am I used to") and the short-term average reinforcement signal ("how am I doing now") (cf. Schweighofer & Doya, 2003). Artificial affect is a measure for how good or bad the situation of the agent is.

In this thesis we explore, among other things, how affect can be used to influence learning by controlling when to *explore* versus *exploit*. As mentioned earlier, animals need to sometimes explore their environment, sometimes exploit the knowledge they have of that environment. Simulated animals also need to do so. To learn where the cheese is, learn different routes to the cheese, learn alternative cheese locations, adapt to new cheese locations, etc., a simulated mouse sometimes needs to explore (to find new stuff) and sometimes needs to exploit (to eat cheese). Controlling exploration versus exploitation is an important problem in the robot learning domain. By using artificial affect to control exploration, and by coupling artificial affect to affect in the psychological literature, an important step is made towards autonomous control of learning behavior in a way compatible with nature. We show that in some cases it is indeed beneficial [1] to the learning simulated animal to control exploration and exploitation by means of artificial affect.

A second aspect explored in this thesis is how affect can be used to control learning more directly, much like a parent that approves or disapproves of a child's behavior. We study, using a simulated robot, the effect of a human observer parenting a robot "child". The robot has to learn a certain task, and the human observer can approve or disapprove the robot's actions by expressing emotional expressions to a camera. The expressions are analyzed in terms of positive and negative affect and fed to the learning robot. This reinforcement signal is used to train the robot, in addition to the normal reinforcement signals given to the robot by the environment it behaves in. We show that learning can improve[2] if such social-based feedback is added to the learning mechanism.

1.2 Computational Models, Psychology and Artificial Intelligence.

Before entering the specifics of the research described in this thesis, a short introduction into the relation between computational models, psychology and artificial intelligence is useful. Computers can be used to model many different phenomena and systems. For example, weather forecasts in fact result from computational (mathematical) models that simulate interaction patterns between the different elements that constitute "the weather", such as air pressure, wind speeds, land elevation, etc. So in essence, a weather forecast is a prediction of the "theory of the weather" by means of a computational model of that theory. In the same spirit, computational models exist that are inspired by, based on, or explicitly implementing psychological theories. Depending on the level of fidelity to the theory, the model can be used to gain insights into, and potentially predict consequences of the psychological theory.

On the other hand, natural theories (such as psychological, economical and biological ones), once implemented, can be very useful in the computer science domain itself. Consider, for example, the Traveling Salesman Problem (TSP), a

[1] Beneficial in terms of (1) effort involved (steps) in finding solutions, and (2) more rewarding solutions.

[2] Improvement in terms of quicker learning of the solution to the task at hand.

typical computational problem defined by finding the shortest route (or at least a route shorter than an arbitrary given length *K*) that visits all locations from a set of locations exactly once (e.g., a traveling salesman that wants to travel from city to city in the most efficient way). TSP is an *NP*-complete problem. In short, this means that to check *if* a given route is a solution to a certain instance of the TSP problem (meaning that the route addresses all locations and is shorter than length *K*), a polynomial number of calculations is needed[3]. Checking a solution is easy in terms of time needed for checking. However, *finding* the shortest route (or deciding if a route shorter than *K* exists) generally takes an exponential amount of calculations, so finding the best route is difficult. This is due to the fact that the number of possible routes that exist between a set of locations grows exponentially with the number of locations. The number of possible routes becomes extremely large even for a small number of locations. An exact solution (i.e., the best route) to this problem is often unnecessary for a real salesman, and for large sets of locations practically impossible. Biologists have studied the behavior of ants intensively and found that ants have an interesting way to find shortest routes to food by leaving scent trails that grow stronger every time an ant uses the same route and finds food at the end. By doing so, ant colonies as a whole have evolved a practical, approximate solution (a.k.a. *heuristic)* to the problem of finding shortest paths. Currently, much research is being done on ant-colony-based heuristics to find practical solutions to, e.g., the Traveling Salesman Problem (Dorigo & Stützle, 2004). This example shows that natural theories can inspire the search for solutions to problems in computer science.

Computational models can thus be used to simulate real-world phenomena, and theories about the real-world can inspire the search for solutions to computer science problems, a notion underlying natural computing in general (Rozenberg & Spaink, 2002). Let's specifically look at the role of computational models in psychology, as well as the role of psychology in computer science.

[3] Polynomial in this context means that the number of calculations needed is expressible in terms of a power over the size of the problem. So, given *n* locations, checking if a route addresses all locations could take, e.g., n^2 calculations, denoted as $O(n^2)$, the complexity *order* is called quadratic. Note that for TSP, there are representations of the problem for which the order for checking a solution is actually $O(n)$: compare if the route contains all *n* locations; sum over all route's segments to obtain the route's length *L* and compare if $L < K$. Note also that the size of a TSP instance is not measured in terms of the number of locations, but in terms of the number of possible location transitions (the potential to move from one city to another); it is not relevant to the complexity of the problem how many locations there are, but in how many ways one can address them all. A polynomial number of calculations is assumed to be *tractable* ("easy" to solve), while an exponential number of calculations (expressible as an exponent, not as a power) is *intractable* ("hard" to solve).

Psychological theories often establish relations (correlations, effects, causality) between different aspects of the human mind and observable behavior. Such relations are often found using sophisticated psychological tests that measure the relation between different *constructs*. A construct is a measurable theoretical abstraction for a certain characteristic, e.g., the construct "intelligence" measured with an IQ test representing the level of non-specific skills a person has. Relations between constructs can be shown in different ways. Most commonly used are the experimental approach aimed at:

- causality; measure construct *A*, do something to construct *B*, than measure construct *A* again to find out if *B* influenced *A* in some way,
- correlation; measure both *A* and *B* at the same time and try to find a correlation between both, and
- longitudinal effects; measure *A* at intervals for a period of many years, manipulate *B*, and try to find trends in *A* over time.

Of course, these approaches exist with or without control groups, with or without blind and double blind setups, and so on.

Aimed at understanding the human mind, psychologists want to study not only relations between constructs but also want to understand the mechanisms responsible for these relations; a notoriously difficult goal, as experimenters cannot look in detail in a persons head. Clever experiment designs have by now been developed that aim at looking into the mind. An impressive example of this can be found in the cognitive psychology domain, e.g., in the domain of working memory and attention. To investigate a relatively simple question such as "can a person attend to, and process two different stimuli at the same time", extremely complex experiment designs have been developed to answer it; not because this is fun, but because the answer must be interpretable in terms of an underlying mechanism. In concrete terms this means that, if the answer is, for example, "yes, persons can do that", the following questions immediately pop up. How many tasks can we simultaneously execute? What task-load is permissible? What if one of the tasks is a heavy one and the other is not, and would performance on the latter be compromised? What if one of the tasks is personally relevant? What if one of the tasks was a task the person is trained on, and to what extent can tasks be executed simultaneously under the assumption that they are indeed trained? How much training is needed? These questions are not so much questions about relations anymore, but in fact questions about mechanisms such as "how does working memory capacity function?", "how is context switching executed by the human brain?", and "how do we concentrate (what *is* concentration)?". The experiment designs needed to study such questions are extremely complex, and very hard to grasp in terms of their consequences for the conclusions (e.g., didn't we forget to control for this or that phenomenon). This is what makes experimental psychology such a difficult and challenging scientific enterprise, for which strong research methods, many different theories and exact reporting of results are critical.

Fortunately (especially for computer science graduates with a strong interest in psychology in search for a topic for their PhD thesis), psychology has added a new type of experiment to their research weapon arsenal, a weapon specifically targeted at understanding mechanism: computer simulation. Computational models need to be specified at a detailed level. As such, in order for a model to execute, mechanism details have to be filled in. If this filling in is done based on a psychological theory, the model becomes a more detailed version of that theory. By executing a computational model, it can provide insights into possible mechanisms underlying the relations between constructs. More importantly, if a psychological theory already proposes potential mechanisms, the computational model can predict consequences of these mechanisms, thereby helping to refine the theory.

Interesting examples include neural network models of human working memory and attention (Dehaene, Sergent & Changeux, 2003), but also the many computational models of emotion based on cognitive appraisal theory that have been implemented in computer systems. Cognitive appraisal theory assumes that emotions result from an individual's cognitive evaluation of the current situation in terms of his or her goals and knowledge. Evaluation is often assumed to be symbol manipulation. As computers are good at such systematic symbol manipulation, this type of theory has been immensely popular as basis for computational models of emotion in (simulated) robots. The development of computational models based on cognitive appraisal theory advances cognitive appraisal theory by refining them (Broekens & DeGroot, 2006; Wehrle & Scherer, 2001). Assumptions in the theory need to be made explicit when used in a computer program.

On the one hand, computational modeling is useful to psychology, while on the other, as we will see now, psychology is useful to computer science, most notably to the field of artificial intelligence.

Broadly speaking, *Artificial Intelligence* (AI) (Russell & Norvig, 2003) studies how computer programs can solve problems, inspired by how nature (including animals, cells, molecules, etc.) solves problems. Intelligence in AI is a vast concept. It includes reactive behavior of autonomous robots aimed at solving concrete problems (e.g., simulated ants in the traveling salesman problem heuristic mentioned above), adaptive stock-price prediction software, and symbolic reasoning processes aimed at transport and military operations planning. In AI, a computer program (the mechanism used to simulate nature) is also defined in a broad way. A program in AI can range from a collection of preprogrammed algorithms that execute planning routines to find optimal planning solutions in advance (e.g., planning an optimal route for a transport company), to reward-based learning mechanisms that continuously adapt their input-output behavior such that the robot they are controlling is able to learn new tasks. So, AI is not exclusively about robots, nor is every robot intelligent. AI is not exclusively about putting loads of knowledge in a database and programming an algorithm that reasons over that knowledge, nor is every knowledge base intelligent. And, to do away with another common misconception: the grand aim of Artificial Intelligence is not about creating intelligence that is artificial as in “fake”, “dumber than real”, and “superficial”, it is about studying the processes and mechanisms of intelligence using artificial means, such as digital computers. If there is a common grand “creational” aim then this would be to develop intelligent, autonomous systems that are able to think and act for themselves, in a way that reflects the wit and cunning of natural intelligence.

Many of the techniques used in AI directly come from other disciplines, such as neuroscience, psychology and biology. For example, artificial neural networks are based on the work by the neuropsychologist Donald Hebb (1904-1985), who described the learning process of neurons in terms of the correlation between pre- and post-synaptic firing, now called *Hebbian learning*. If two neurons are connected through a synapse, and both the pre-synaptic neuron A (exciting neuron B) and the post-synaptic neuron B (excited *by* A) activate (fire) at about the same time, the strength of the connection is increased, thereby increasing the probability that neuron A excites B in the future. This model underlies many of the learning mechanisms implemented in artificial neural networks, but also underlies connectionist learning models in general.

Another, more specific, example is the application of *Soar* in the area of computer games research as well as medical image analysis. Soar (originally for State, Operator And Result) is a cognitive architecture aimed at problem solving through rule matching. It is based upon the idea of a unified theory of cognition, proposed by Newell (1990), integrating theories of cognition from many different disciplines. Key elements of Soar are its ability to plan for, reason about and act upon a situation using rule matching in recursive thought cycles. In every cycle, all rules that apply to the current situation activate. The activation strength of a rule depends on how well the rule matches the current situation. The most strongly activated rules are allowed to propose new “facts”, such as actions that can be executed by the robot controlled by the Soar program. If no rules activate based on the current situation, a new “problem” is created, and Soar tries to recursively solve this problem. Once the problem is solved, Soar creates a new rule for future use, solving that problem more efficiently should it pose itself again. This architecture, proposed as a symbolic theory of cognition, has been used to build intelligent computer game agents that predict what other agents (e.g., the user) will do (Laird, 2001). In the medical domain it is currently being used in image analysis software agents: specialized programs responsible for analyzing a specific type of information in an image to coordinate, e.g., analysis of coronary plaque images (Bovenkamp et al., 2003).

We have seen that computer science—specifically artificial intelligence—and psychology—specifically cognitive psychology—are fields that strongly influence each other in many ways. This influence dates from the very early 1950’s. Alan Turing’s (1950) well-known paper on machine intelligence was published in *Mind*, a psychological and philosophical journal, at about the same time as the seminal papers that started the cognitive revolution in psychology. Donald Hebb (1949) presented such a clear description of how brains learn that this opened up an information processing view of the mind. The mechanisms he described have by now been applied in robotics and AI many times.

Most important to this thesis are the concepts *affect* and *instrumental conditioning*. Instrumental conditioning underlies Reinforcement Learning (RL) (Sutton & Barto, 1998), a method that has proven to be critical for artificial task-learning. As we have used RL as a model for learning in our research, it is one of the cornerstones of our approach. We devote the next section to it. We use artificial affect to influence learning. Therefore, affect is the second cornerstone. We devote Section 1.4 and Chapter 2 to the latter topic.

1.3 Learning, Instrumental Conditioning, Reinforcement Learning.

Animals learn behavior in a variety of ways, such as by imitation, by play, and by trial and error. Instrumental conditioning is the more formal name for learning behavior by trial and error. For example, rats learn to push buttons or pull levers in order to receive food. To learn this behavior they have to try actions *before* they know the result of that action. It could be that pushing a button results in the rat being punished. As there is no way to know this beforehand, the rat has to try to push the button, at least for the first time. After pushing it, the rat either receives food, or some kind of punishment (e.g., a loud sound). The animal learns to repeat the actions that lead to food, and avoid actions that lead to punishment.

This is called instrumental conditioning (see Anderson, 1995): learning to repeat or avoid actions in a certain situation, based on reward and punishment.

Interestingly, many animals learn to execute sequences of actions. To take our rat example, the rat not only learns to push the button for food, it also learns to walk to the button *after* having looked around for the button *after* having entered the specific rat-maze room in which the button is located, etc. By reinforcing a certain situation-action couple, not only the last action is influenced, but also the sequence of environment-rat interactions leading to that reinforcement. Further, this sequence is better learned if it is repeated. So, repetition of a sequence of interactions ending with reinforcement enables the rat to learn that sequence better and better. The same mechanisms can account for many goal-directed behaviors of humans. We rarely do something without having received rewards, and by training we become better at it. Sometimes the reward is indirect, such as in the case of money. It is straightforward to argue that money has become a reinforcer by itself because humans have associated it with more natural reinforcers (Anderson, 1995), such as food (restaurants, candy), play (vacation, toys) and social interaction (having a drink with friends, going to the theatre or a rock concert, distributing candy at school). We learn to work (a long sequence of actions) for money, because money gives us naturally reinforcing stuff.

Finally, *discounting* is a concept of critical importance: rewards and punishments in the future are perceived as less important than in the here and now. Animals discount the value of reinforcement, dependent upon the time passed between administration of the reinforcement and the action to be reinforced. As a result, reinforcement most strongly influences the action executed just before receiving the reinforcement.

In this section we will see that the machine learning concept of Reinforcement Learning is a very good model for instrumental conditioning.

1.3.1 Reinforcement Learning

Strongly related to instrumental conditioning, there is a form of machine learning called *Reinforcement Learning*. Reinforcement Learning (RL) (Sutton & Barto, 1998) is a computational framework describing how in an environment appropriate actions can be learned purely based on exploration and reinforcement. Actions are appropriate if they maximize some signal from the environment, say a reward. As such, RL, is a particular computational model of instrumental conditioning[4]. A formal description of RL is the problem of learning a function

[4] Dayan (2001) and Kaelbling, Littman and Moore (1996) discuss some of its limitations.

that maps a state to an action, such that, given a certain history of state-action transitions, for all states this mapping results in an action that yields the highest cumulative future reward as predicted by that history of state-action transitions. In normal language this means that RL attempts to recognize the best possible action in a situation, given a certain amount of experience.

We have used Reinforcement Learning as a basis for learning in this thesis. The main reason for this choice is that RL maps very well to animal task learning (instrumental conditioning). The second reason is that RL has proven to be the most successful paradigm for the machine learning of tasks composed of multiple actions that are not known in advance. Other forms of learning, such as supervised learning, need a human observer. RL does not, it learns by trial and error, providing a clear benefit: a RL system learns autonomously. This is important for, e.g., robot learning. By investigating the relation between RL and affect, we hope to advance a well known machine learning paradigm as well as shed some light on the potential relation between affect and learning.

In essence, RL aims at solving the *credit assignment* problem (Kaelbling, Littman & Moore, 1996). That is, how much credit should an action get, based on its responsibility for receiving current and future rewards; in other words, how should an action in a certain situation be valued given its immediate reward as well as all rewards that might follow? Note that from now on we will talk about *reward* when we mean reinforcement. Reward can thus be positive and negative. A classical representation of a function that represents a solved credit assignment problem is a 2-dimensional table with cells representing the value of all actions in all possible states, rows representing states, and columns representing actions (Table 1.1). If this table is used for control, i.e., to select actions for execution by a simulated animal, the current observed state is used as row entry, and the action belonging to the cell with the highest value on that row is selected. For example, if the simulated animal would be in state *choice* (Figure 1.1), the best action to perform is *left*. When the action *left* is executed a state change occurs, and the next state is observed by the simulated animal; *food* in our case. Now the process of action-selection can be repeated.

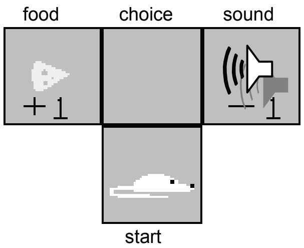

Figure 1.1 The maze solved by the function depicted in Table 1.1. The cheese has a reward of +1, while the loud sound has a reward of −1. States are called food, choice, sound, start for “mouse at cheese”, “mouse at junction”, “mouse in sound room”, and “mouse at start”. This is a four-state problem.

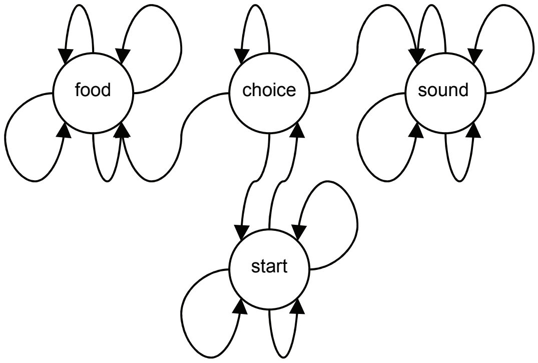

Figure 1.2 A state-transition diagram for the maze presented in Figure 1.1. Arrows denote move actions. Probabilities are assumed to be equal to 1 (i.e., choosing, for example, *up* always results in the state pointed to by the up-arrow). The states, *sound* and *food* are terminal states.

	left	*right*	*up*	*down*	*eat*
start	0.125	0.125	0.25	0.125	0
choice	0.5	−0.5	0.25	0.125	0
food	0	0	0	0	1
sound	−1	−1	−1	−1	−1

Table 1.1 The classical representation of a function that solves a specific credit assignment problem, in our case food-finding in a simple maze (Figure 1.1). The discount factor, γ, equals 0.5. So the importance of future rewards drops with a factor of 2 for every step in between an action and a reward. States are called *food*, *choice*, *sound*, *start* for "mouse at food", "mouse at junction", "mouse in sound room", and "mouse at start" respectively. We assume that when the mouse arrives at food or sound, it can not exit that place by itself. We further assume that moving outside the maze does not result in a state change. This table presents the solution to our four-state problem.

At an architectural level, the RL problem can be formally described as follows. It consists of a set of states, S, a set of actions, A and a transition function $T : S \times A \times S \rightarrow [0,1]$ defining how the world changes under the influence of actions giving the probability $T(s, a, s')$ that action a in state s results in state s', where the sum over all s' of $T(s, a, s')$ equals 1. Further, a reward function $R : S \times A \rightarrow \Re$ and a value function $V : S \rightarrow \Re$ are defined. The states S contain representations of the world perceived by the *agent*, such as a *start* state, a *food* state etc. Note that from now on we use the term *agent* to refer to a simulated animal or robot. The actions A contain all possible actions the agent can execute, such as *left*, *right*, *up*, *down* and *eat*. The transition function defines the probability of ending up in one state, assuming a current state, s and action, a. So, the agent's world is probabilistic[5]. The reward function defines the reward for a certain action, a, when executed in state, s. The value function maps a state, s, to a cumulative future reward. So, if an agent knows T and V the optimal next action can be selected using:

[5] but stationary, i.e., the probabilities do not change (Kaelbling et al., 1996).

$$a^* = \arg\max_a \left(R(s,a) + \gamma \sum_{s' \in S} T(s,a,s')V(s') \right), \text{ with } \gamma \text{ the discount factor (1.1)}$$

The best action a^* is the action with the highest sum of immediate reward $R(s, a)$ and value predictions $V(s')$, over all possible next states s' resulting from action a^*, weighted according to their probability of occurrence $T(s ,a ,s')$. Note that the summation in formula (1.1) is needed as in a probabilistic world multiple states s' might result from action a. In our example, moving *up* in state *start* would be the best action, because $R(start, up) + 0.5T(start, up, choice)V(choice) = 0 + 0.5*1*0.5 = 0.25$, which is the highest value (we assume that the probability of ending up in state *choice* after executing *up* in state *start* equals 1, so in our case we only have one possible next state s' after executing action *up* in state s). However, to select this action we have to know both $V(choice)$ and $T(start, up, choice)$.

Solving the credit assignment problem has thus become a question of learning the value function V, together with the transition function T. The main question is, how? The short answer is: by trial and error; try actions in states, record the received reward and the resulting state, and update both V according to the reward, as well as T according to the probability of arriving in that new state. The longer, formal answer is: by value propagation according to the following formula:

$$V(s) \leftarrow \max_a \left(R(s,a) + \gamma \sum_{s' \in S} T(s,a,s')V(s') \right) \qquad (1.2)$$

which is equivalent to $V(s) \leftarrow val(a^*)$, with $val(a^*)$ the value of action a^*

The formula updates the value for state s with the immediate reward $R(s, a)$ and discounted future values $V(s')$ for all s' possibly resulting from action a weighted according to the probability $T(s, a, s')$ that transition $s \rightarrow s'$ occurs due to action a. Again, action a is chosen such that it is the best one possible. This enforces conversion of values to the highest possible value attainable by the agent.

By now, many different version of RL exist that all solve the credit assignment problem in a slightly different way (for a dated but excellently written overview, see Kaelbling et al., 1996). In general there are two different types of RL approaches; *model-based* and *model-free*. Model-based approaches have (or learn) a model of the world that consists of a (probabilistic) state transition structure (Figure 1.2). Model-based approaches thus have a function T. The research in Chapter 3 and 4 is based on model-based RL. Model-free approaches do not have such a world model. Model-free approaches thus need to learn *V* in a different way, as they do not possess the function *T* while this function is needed for value propagation as described in formula (1.2). The research in Chapter 6 is based on model-free RL.

In the model-free case, *V* can be learned in the following way. It can be shown (Singh, 1993) that the following formula converges to an optimal value function *V*, if a sufficient and unbiased amount of exploration occurs during learning, and the learning rate *α* is gradually decreased from 1 (in the beginning of learning) to 0 (at the end of learning):

$$V(s) \leftarrow V(s) + \alpha\big(r + \gamma V(s') - V(s)\big), \text{ with } r \text{ the reward} \qquad (1.3)$$

It is quite well possible to intuitively grasp this without proof. If an agent has an infinite amount of time to keep trying things in a world, it eventually bumps infinitely many times into all possible situations that exist in that world. This means that it will see the transitions *s*→*s'*,*s''*,… for all *s* many times. Every such transition updates *V*(*s*) a little bit, so together *V*(*s*) accumulates the results of all these transitions. It correctly estimates the value of *s* by sampling a representative number of transitions resulting from *s*. So, an agent (or real animal for that matter) has to explore—i.e., sample a representative number from all possible interactions with the environment—to be able to learn a useful value function. After exploration, the agent can use the learned value function to act, i.e., the agent can exploit its knowledge. The *exploration – exploitation tradeoff* is a very important issue in Reinforcement Learning (Sutton & Barto, 1998). Without a good mechanism to decide when to explore versus exploit, RL cannot learn an optimal value function.

It is important to note here that there are ways in which an artificial agent can learn an optimal interaction model (in terms of maximizing cumulative reward). One of these is to let the agent first explore a large amount of time, and then switch to an exploitation mode. However, this is not plausible from a natural point of view. No animal can afford to purely explore, as this is just too risky. In our learning models (Chapter 3 to 6), we take this into account. We have no separate exploration – exploitation phases; our agents learn the value and transition function while at the same time using these for action selection (called *certainty equivalence*, see Kaebling et al, 1996). Our agents thus assume that their world model is a correct estimation of the world they interact with.

1.3.2 Reinforcement Learning as a Model for Instrumental Conditioning

As mentioned before, one of the main reasons for using Reinforcement Learning (RL) as learning mechanism in studying the interplay between affect and learning is that RL very well models instrumental conditioning. RL models instrumental conditioning in at least three important ways.

- First, it associates rewards with the probability of execution of actions in a certain situation, as in instrumental conditioning. The simulated animal learns to repeat actions based on an association between reward and action.
- Second, by repetition the learned association becomes more accurate, and as such the probability to execute actions that result in reward becomes larger (positive reward) or smaller (no reward, or punishment).
- Third, the learned value for a situation can influence the execution of actions in earlier situations. We thus see that RL provides an answer to how sequences of actions can be learned by trial and error: propagate the reward through the sequence back to the beginning such that the right amount of credit is given to the individual actions in the sequence.

Recently, the mechanism of Reinforcement Learning has been tied to neural substrates involved in instrumental conditioning. For example, there are strong links between dopamine brain systems and RL (Dayan & Balleine, 2002; Montague, Hyman & Cohen, 2004; Schultz, Dayan & Montague, 1997). It seems that neurons in these regions encode for the RL *error signal*, i.e., the change to the expected value of a situation, $\Delta V(s)$. More recently Foster and Wilson (2006) showed that awake mice replay in reverse order behavioral sequences that led to a food location; a crucial finding for the above mentioned link. It suggests that mice can replay sequences backward from the goal location to the start location. This is a mechanism that would be needed to speed up value propagation back to the beginning, and is highly compatible with the RL concept of *eligibility traces* (Foster & Wilson, 2006). An eligibility trace (for details see Sutton & Barto, 1998) is a state sequence leading to a certain reward or punishment. In RL, eligibility traces can be used to speed up learning. The idea is to update the complete sequence based on that reward (such a sequence represents a trace of situations that is eligible for the resulting reward). In RL, updating the value of states in this trace can be done in any order. In nature, backwards is more plausible than forwards for the following reason. Assumed that the brain is a connectionist architecture primarily learning by means of Hebbian mechanisms, in order for two situation representations to transfer a characteristic (e.g., reward) between each other, both have to be active at the same time. If a state sequence is replayed backwards, pair-wise activation of two consecutive states, for all states in the sequence starting at the end, would in principle suffice to (partly) transfer the reward to the start of the sequence. However, for any other order to get the same value propagation result, it would need either massive repetition of activated pairs of representations or activation of all pairs at the same time. So, activation of the state sequence from the end, back to the beginning seems more efficient than any other order[6]. It is therefore interesting to see that mice seem to indeed replay *in reverse order* the "states" they visited while walking towards the food.

Finally, animal learning by trial and error closely matches RL in how experience of the world is built up: by means of a sufficient number of interaction samples to build up the value function. Trials are samples from all possible interactions with the environment; errors (rewards) change the value and reward functions learned by the animal. If an animal is a good explorer, it will be better at finding optimal solutions because it samples more possibilities from the environment, therefore the animal's resulting value function has more chance to better estimate the real value function. On the other hand, exploration is risky: if you don't know what the result will be, you could die. Animals that do not explore will stick to their current interaction pattern. This means that as long as the interaction pattern is appropriate for the environment they are in, they will do better than explorers: they don't waste time exploring useless options while they have a good option available. However, as soon as the environment changes, they will die because of the useless option and the lack of exploration. To learn a good value function, a sufficient amount of exploration is needed. So, also in real life, the tradeoff between exploration and exploitation is important. Actually it is much more important in real life, as one stupid action can result in death or illness, while in a simulated world it only results in a negative reward. A second difference is that in real life one can not afford to have a pure exploration phase: this would most certainly result in at least one very stupid action, hence death. As a result, the exploration – exploitation tradeoff is even more important. Both have to be in balance for an agent to survive. In Chapter 3 and 4 we explore to what extent artificial affect can be used to control the exploration - exploitation tradeoff. We have based these studies on how affect influences learning in humans, a topic introduced in the next section, and in more detail in Chapter 2. In order to stay consistent with nature, we do not separate exploration - exploitation phases.

Although from this description it seems that RL has been used primarily to simulate learning animals, this is not the case. RL has been widely used to learn computers to play games (e.g., Tesauro, 1994), to control cars to autonomously drive based on visual input (e.g., Krödel & Kuhnert, 2002) and to control robots (e.g., Theocharous, Rohanimanesh & Mahadevan, 2001).

[6] Interestingly, value propagation in RL is in the same direction, that is, backwards.

1.4 Emotion, Affect and Learning

In this thesis we specifically focus on the influence of affect on learning. Affect and emotion are concepts that lack a single concise definition, instead there are many (Picard et al., 2004). Therefore we first explain the meaning we will use for these terms. In general, the term emotion refers to a set of in animals naturally occurring phenomena including motivation, emotional actions such as fight or flight behavior and a tendency to act. In most social animals facial expressions are also included in the set of phenomena, and—at least in humans—feelings and cognitive appraisal are too (see, e.g., Scherer, 2001). A particular emotional state is the activation of a set of instances of these phenomena, e.g., *angry* involves a tendency to fight, a typical facial expression, a typical negative feeling, etc. Time is another important aspect in this context. A short term (intense, object directed) emotional state is often called an *emotion*; while a longer term (less intense, non-object directed) emotional state is referred to as *mood*. The direction of the emotional state, either positive or negative, is referred to as *affect* (e.g., Russell, 2003). Affect is often differentiated into two orthogonal (independent) variables: *valence*, a.k.a. pleasure, and *arousal* (Dreisback & Goschke, 2004; Russell, 2003). Valence refers to the positive versus negative aspect of an emotional state. Arousal refers to an organism's level of activation during that state, i.e., physical readiness. For example, a car that passes you in a dangerous manner on the freeway, immediately (*time*) elicits a strongly negative and highly arousing (*affect*) emotional state that includes the expression of anger and fear, feelings of anger and fear, and intense cognitive appraisal about what could have gone wrong. On the contrary, learning that one has missed the opportunity to meet an old friend involves cognitive appraisal that can negatively influence (*affect*) a person's mood for a whole day (*time*), even though the associated emotion is not necessarily arousing (*affect*). Eating a piece of pie is a more positive and biochemical example. This is a bodily, emotion-eliciting event resulting in mid-term moderately-positive affect. Eating pie can make a person happy by, e.g., triggering fatty-substance and sugar-receptor cells in the mouth. The resulting positive feeling is not of particularly strong intensity and certainly does not involve particularly high or low arousal, but might last for several hours.

We use affect to denote the *positiveness* versus *negativeness* of a situation. In the studies reported upon in this thesis we ignore the arousal a certain situation might bring. As such, positive affect characterizes a situation as good, while negative affect characterizes that situation as bad (e.g., Russell, 2003).

Emotion plays an important role in thinking, and evidence is abundantly available. Evidence ranging from philosophy (Griffith, 1999) through cognitive psychology (Frijda, Manstead & Bem, 2000) to cognitive neuroscience (Damasio, 1994; Davidson, 2000) and behavioral neuroscience (Berridge, 2003; Rolls, 2000) shows that emotion is both constructive and destructive for a wide variety of behaviors. Normal emotional functioning appears to be necessary for normal behavior.

Emotion[7] influences thought and behavior in many ways. Emotion can be a motivation for behavior. Emotion is related to the urge to act (e.g., Frijda & Mesquita, 2000): run away when in danger, fight when trapped, laugh and play when happy. Specific emotions trigger specific behaviors (e.g., fight or flight). So, emotion is not only related to the *urge* to act, some emotions—when strong enough—make us really act.

Emotion and feelings influence how we interpret stimuli, how we evaluate thoughts while solving a problem (Damasio, 1996) and how we remember things. A person's belief about something is updated according to emotions: the current emotion is used as information about the perceived object (Clore & Gasper, 2000; Forgas, 2000), and emotion is used to make the belief resistant to change (Frijda & Mesquita, 2000). Ergo, emotions are “at the heart of what beliefs are about” (Frijda et al., 2000). As shown by the “should I stay or should I go” scenario presented earlier in this introduction, we often decide to do something based on how that option feels to us.

Finally, emotion influences information processing in humans; positive affect facilitates top-down, “big-picture” heuristic processing while negative affect facilitates bottom-up, “stimulus analysis” oriented processing (Ashby, Isen & Turken, 1999; Gasper & Clore, 2002; Forgas, 2000; Phaf & Rotteveel, 2005). As a result, positive affect relates to a “forest” or goal-oriented look (we interpret what we see in the context of our existing knowledge), while negative affect relates to a “trees” or exploratory look (we critically examine incoming stimuli as they are).

Several psychological studies support that enhanced learning is related to positive affect (Dreisbach & Goschke, 2004). Others show that enhanced learning is related to neutral affect (Rose, Futterweit & Jankowski, 1999), or to both (Craig, Graesser, Sullins & Gholson, 2004). Although much research is currently being carried out, it is not yet clear how affect is related to learning in detail.

In this thesis we computationally address this issue: in what ways can affect influence learning. We do not model categories of emotions nor use emotions as

[7] An emotion is different from a feeling. A feeling is in essence your mental representation of yourself having the emotion.

information in symbolic-like reasoning. So the research goal has not been to investigate how agents can reason "emotionally", such as in the work by Marsella and Gratch (2001), or interact emotionally with humans (Heylen et al, 2003).

1.5 Questions Addressed and Thesis Outline.

To study the influence of affect on learning, in a Reinforcement Learning setting, we first have to evaluate whether affect can be used in this context: we have to define affect in a Reinforcement Learning context. In Chapter 2 we define artificial affect in detail. In Chapter 3 to 6 we study three different ways in which affect can influence learning, where learning in each chapter is modeled using a different variation of RL.

In Chapter 3 we investigate how artificial affect can control exploration versus exploitation. As the amount of exploration strongly influences learning behavior, and as it has been found (e.g., in the studies mentioned earlier) that affect relates to broad (explore) versus narrow information (exploit, goal directed) processing, we have investigated how artificial affect can control exploration versus exploitation in agents. A simulated "mouse" in a grid-world maze can either search for "cheese" (eating cheese is its goal) by trying actions it does not know the consequences for (explore), or use its model of the environment it has built up so far in an attempt to walk to the cheese by trying actions it thinks it knows the consequences for (exploit). We couple artificial affect to exploration and exploitation in different ways, according to studies reported by Dreisbach & Goschke (2004) and Rose et al. (1999): positive affect increases exploration (and negative affect increases exploitation) and vice versa. In RL terms, we use artificial affect as meta-learning parameter (see also Doya, 2002) to control exploration versus exploitation by dynamically coupling it to the greediness of the *action-selection* function responsible for making this choice (the β parameter of the Boltzmann distribution, in our case). A meta-learning parameter is a parameter that influences learning, but does not contain information about the task to be learned per se, e.g., the choice to explore versus exploit, or the speed with which to forget knowledge you had acquired. We use a version of RL that is similar to *Sarsa* (Rummery & Niranjan, 1994; Sutton, 1996). The main findings are that (1) both negative affect and positive affect can be beneficial to learning, and (2) negative affect seems to be related to less selective decisions while positive affect is related to more selective decisions.

In Chapter 4, we investigate the influence of affect on thought. Instead of studying the influence of artificial affect on action-selection in a purely reactive agent, we now study the influence of artificial affect on "thought selection" in a more cognitive agent. In our study, we have defined thought as internal simulation of potential behavior, according to the S*imulation Hypothesis*, proposed by Hesslow (2002) and Cotterill (2001). This process of simulation uses the same brain mechanisms as those used for actual behavior. For example, if I consciously think of going home and play games, I, in a sense, go home and do so without moving my body. Simulating going home thus enables me to evaluate how I feel about going home by triggering the same brain areas and processes that would have been triggered if I went home and started playing. This again enables me to decide whether I should do it or not, showing that simulation could be useful for decision making and action selection. We have developed a variation to the model-based RL paradigm, called *Hierarchical State Reinforcement Learning*, which enables us to study this question. We computationally investigate, again using a grid-world setup, the influence on learning efficiency when artificial affect controls the amount of internal simulation. Artificial affect is dynamically coupled to the greediness of the *simulation-selection* mechanism responsible for selecting potential actions for internal simulation. As such we model affective modulation of the amount of thought during a learning process. The main findings are that (1) internal simulation has an adaptive benefit and (2) affective control reduces the amount of simulation needed for this benefit. This is specifically the case if positive affect decreases the amount of simulation towards simulating the best potential next action, while negative affect increases the amount of simulation towards simulating all potential next actions. Thus, agents “feeling positive” can think ahead in a narrow sense and free-up working memory resources, while agents “feeling negative” are better off thinking ahead in a broad sense and maximize usage of working memory.

In Chapter 5 we discuss related and future work in the context of the studies presented in Chapter 3 and 4.

In Chapter 6, we investigate how affect can be used to influence behavior of others. Emotion and affect are important social phenomena. One way in which affect is important socially is that it enables effective parenting. Affect communicated by a parent can be seen as a reinforcement signal to a child. In this chapter we investigate the influence of affect communicated through facial expressions by a human observer on learning behavior of a simulated “child”. We thus investigate the effect of parenting a simulated robot using affective communication. Two important differences exist between the study in this chapter and those in Chapters 3 and 4. First, we use a continuous (non-discrete) grid-world setup, use real-time interaction between the robot and the human “parent”, and use a specifically developed neural-network approach to Reinforcement Learning applicable to this context. This has been done to match real-world learning problems more closely. Second, we use affect in a different way. In Chapter 3 and 4, we use artificial affect as defined in Chapter 2; i.e., a long-term signal originating from the simulated agent, used by the simulated agent to control its own learning-parameters. In contrast, in the experiments reported in Chapter 6, we use affect as a short-term signal related to emotion, originating from an observing "parent" agent, used to influence the reinforcement signal received by the simulated robot. The main finding is that the simulated robot indeed learns to solve its task significantly faster (measured quantitatively) when it is allowed to use the social reinforcement signal from the human observer. As such, this chapter presents objective support for the viability and potential of human-mediated robot-learning.

In Chapter 7, we take a theoretical approach towards computational modeling of emotion. We present a formal way in which emotion theories can be described and compared with the computational models based upon them. We apply this formal notation to cognitive appraisal theory, a family of cognitive theories of emotion, and show how the formal notation can help to advance appraisal theory and help to evaluate computational models based on cognitive appraisal theory: the main contributions of this chapter. Although this chapter is quite different from the others, it fits within the general approach: that is, the use of computational models to evaluate emotion theories.

1.6 Publications

A revised version of Chapter 3 has been published in (Broekens, Kosters & Verbeek, 2007). Parts of Chapter 4 have already been published earlier (Broekens, 2005; Broekens & Verbeek, 2005), while Chapter 4 is a slightly revised version of the article by Broekens, Kosters & Verbeek (in press). Chapter 6 has been published in (Broekens & Haazebroek, 2007), while an extended and revised version has been published as a book chapter in (Broekens, 2007). Earlier versions of the work in Chapter 7 have been published (Broekens & DeGroot, 2004c; Broekens & DeGroot, 2006), while a revised version of Chapter 7 is published in (Broekens, Kosters & DeGroot, in press).