Generating and Refining Causal Models for an Emotion-based Agent

Rodrigo Ventura and **Carlos Pinto-Ferreira**
Institute for Systems and Robotics
IST — Torre Norte, piso 6
Av. Rovisco Pais, 1
1049-001 Lisboa
PORTUGAL
{yoda,cpf}@isr.ist.utl.pt

Abstract

Establishing cause-effect relationships is a relevant problem in A.I., particularly for providing autonomous agents with models (for decision making) obtained from their interaction with the environment. The strategy adopted in this paper consists of formulating a predictive model of the relevant consequences of agent actions (or inactions). From the agent perspective, a stimulus is relevant whenever either it is desirable (or undesirable), or when the occurrence of that particular stimulus contributes to the anticipation of desirable (or undesirable) stimuli. The purpose of the agent is to use the formulated causal model to act upon on the environment, in order to avoid undesirable stimuli. To validate the approach, the agent was exposed to a very simple environment. Preliminary results are presented, showing that this approach is worth pursuing.

Introduction

Formulating causal models is one of the first mechanisms underlying intelligent behaviour. To cope with a partially unknown environment, an agent ought to be capable of establishing causal links through interaction with the external milieu. However, a simple mechanism of cause-effect association, based upon the fallacy *post hoc ergo propter hoc,* suffers from the severe disadvantage of being an unsound method of inference. To circumvent this problem, the agent should be prepared to revise beliefs whenever new and contradictory evidence emerges, incorporating it in a refined model. The objective of this research is to investigate how an agent with very little *a priori* knowledge can establish rules concerning the workings of its environment, generating expectations about the future, and being prepared to take adequate courses of action.

The simple emotional agent described in this paper has two diverse and alternating working modes: (i) interacting with the environment, in which the agent stores cases - in the form of sequences ending up in particular situations, and (ii) processing collected cases, in which a causal model is either generated or refined. This agent is based on an architecture already described in (Ventura & Pinto-Ferreira 1998; Ventura, Custódio, & Pinto-Ferreira 1998; 2001; Ventura & Pinto-Ferreira 2002). The key aspects of this approach is the utilisation of a double knowledge representation, a marking mechanism inspired in the work of Antonio Damásio (Damásio 1994), and the introduction of the "desirability vector" which provides an assessment — desirable or undesirable — with respect to each stimulus the agent is exposed to. To bootstrap the agent behaviour, some *a priori* stimuli are associated to particular desirability values. After a process of interaction with the environment, the agent becomes capable of ascribing desirability values to previously "uncoloured" stimuli, so allowing the implementation of a mechanism for expectation generation.

The origins of the interest for studying emotions in AI can be traced back to two main sources: human-computer interaction (HCI) on an affective basis (Picard 1995), and the study of the contribution of emotions to intelligent behaviour (for instance, see (Sloman 1999)). Antonio Damásio has been an influential reference for several researchers (Damásio 1994) in the area of emotion-based agents, namely because the essential role he attributes to emotion as far as rationality is concerned. Related work having Damásio as a fundamental reference was developed, for instance, by Juan Velasquez (Velásquez 1999), which uses the Damásio's idea of somatic marking along with a society of agents approach (Minsky 1988). Another approach based on Damásio's is the one of Gadanho (Gadanho & Hallam 1998), which complements a reinforcement learning architecture with an hormonal system. Several implementations followed another influential reference — the Appraisal Theory of Frijda (Frijda 1986) — for instance, the TABASCO (Staller & Petta 1998), and the SALT&PEPPER (Botelho & Coelho 2001) architectures. Aaron Sloman has been one of the precursors of emotions in AI, defending an architecture based on a reactive, deliberative and meta-management layers (Sloman 1999), where the emotions emerge as alarm systems acting upon one or more layers.

Implementation

The agent is stimulated by the environment with a sequence of symbols. These symbols have no *a priori* meaning, except for one of them — the `X` — which triggers a negative assessment by the agent. We call this negative assessment a negative DV, where DV stands for *desirability vector*. According to the proposed architecture for emotion-based agents, the

desirability vector represents a basic assessment the agent performs about a stimulus, where each component reflects to what degree the stimulus is desirable to the agent, with respect to a specific aspect (Ventura & Pinto-Ferreira 1998; Ventura, Custódio, & Pinto-Ferreira 1998). In the context of this paper, we only consider a single component vector, whose values are either neutral or negative.

The agent begins its interaction cycle with a minimal *a priori* knowledge about the environment, *i.e.,* the negative DV produced by the `X` symbol. Through interaction it formulates and puts a causal model into practice, which relates received stimuli and external actions, with future consequences.

First of all, the agent has to be able to store, in memory, a sequence of the latest symbols to which the agent has been exposed so far. To do so we use a FIFO-like structure which we call "movie-in-the-brain" (MITB). We have explored this concept before (Ventura, Custódio, & Pinto-Ferreira 2001; Ventura & Pinto-Ferreira 2002), where it was used by an agent to store the history of its interaction with the world, in order to learn courses of action to attain desirable states.

Next, the agent needs to collect and store cases, *i.e.,* subsequences of stimuli, which the agent finds relevant to formulate a causal model. Before being exposed for the first time to a `X` symbol, the agent does nothing. When the first `X` symbol appears, the N previous stimuli (present in the MITB) are stored in a database of cases (where N is a parameter — the size of the MITB).

The agent implements two distinct modes of operation: an online mode, where the agent interacts with the environment, collecting cases when appropriate, and acting accordingly with a previously formulated causal model (if any), and an offline mode, where all collected cases are analysed, in order to refine and possibly reformulate a new causal model.

We do not restrict the causal model to a particular technique. In this experiment we use a decision tree structure, using the C4.5 algorithm (Quinlan 1993). However we would like to stress that the model we propose can use any other technique.

To build a decision tree it is not enough to take subsequences leading to negative DV stimuli. The algorithm requires cases which do not lead to a negative DV. This is so because a decision tree partitions the attribute space according to the decision outcomes, requiring the train-set to contain examples associated to all possible outcomes. Therefore, the agent has to be equipped also with the capability of collecting counter-cases, namely negative DV and neutral stimuli. To do so we used the following strategy: when a sub-sequence ending in a negative DV is stored in the database of cases, all symbols found in that sub-sequence are associated with that case. This way, symbols that occur before a negative DV stimulus become associated, each one, with one (or more) cases where they took part. When any one of those symbols is found, the stored case is recalled, compared with the past, and held for tracking. All differences the agent finds between the recalled case and the current one are registered. When the tracking of the recalled case ends, and if any differences were registered, a new case is added to the database of cases.

This database of cases is used next time an offline mode period occurs. When the offline mode occurs for the first time, a brand new decision tree is built. As mentioned, the decision tree is generated using the C4.5 algorithm (Quinlan 1993). The examples used by this algorithm consist of sub-sequences of stimuli. The attributes and values are pairs in the form (n, s) or (n, a), where n is an integer representing the position of the stimulus s and action a in the subsequence. The outcomes are the DV values — negative or neutral — of the final stimuli in the sub-sequences. The final stimulus and action itself are not included in the attributes, since the decision tree is supposed to anticipate the DV *before* it happens. Before finishing an offline mode period, the agent discards the database of cases. For the subsequent offline mode periods, an *ad-hoc* refinement algorithm was used, which will be explained later on this paper.

Once an initial causal model is formulated (a decision tree, in our experiment), the agent can use it to anticipate negative DV stimuli. However, it may happen that the model fails to anticipate correctly a negative DV, or that it anticipates a negative DV that does not follows as expected. In these cases, the model needs to be refined. To accomplish this, these cases where the decision tree fails are added to the database of cases, so that in the next offline period, the agent is able to use them to refine the model.

For the sake of simplicity, we use a simple *ad-hoc* scheme, that works as follows: the agent adds to each leaf[1] the subset of examples (cases) that have led to that outcome. The decision tree refinement is performed using these subsets. The algorithm consists of, for each example, starting at the root, and walking through the tree, until one of the following situations is encountered:

1. At an (attribute) ramification, the corresponding value of the example is not accounted for: in this case, a new leaf is added at this ramification, associating the new attribute value with the example outcome;
2. At an (outcome) leaf, the outcome diverges from the one of the example: a new decision tree is generated, using the C4.5 algorithm, taking the examples stored in that leaf, together with the new example.

A formulated causal model can be used to prevent exposure to negative DV stimuli. The agent is endowed with a built-in behaviour that consists of performing a pre-defined action (symbol `AVOID`) once it anticipates a negative DV for the immediately following stimulus. If that action gives rise to neutral DV, then this corresponds to an unexpected neutral DV. As mentioned before, this originates the storage of a new case, which will be used, in the next offline period, to refine the causal model. In the end, the causal model contains knowledge, not only about relevant stimuli which precedes a negative DV, but also about which actions are capable of preventing negative DV stimuli. This allows the formulation of possible action scenarios. These action scenarios associate courses of action with future consequences,

[1] The ramifications of a decision tree correspond to possible attribute values, and the leafs correspond to possible outcomes.

in terms of the DV, according to the causal model. For instance, simple statements such as "next stimulus has a negative DV!", or "if you perform an `AVOID` action, the next stimulus has neutral DV" illustrate the idea.

Figure 1 shows the architecture of the described agent. During the online mode, stimuli (1) are stored in the "movie-in-the-brain" (MITB). Under certain circumstances, sequences from the MITB are stored (2) and/or tracked (3). During the offline period, a decision tree is constructed or refined (A). The decision tree is used (6) to anticipate (7) what can happen next. This information is used to formulate courses of action (8), and to choose an action to perform (9).

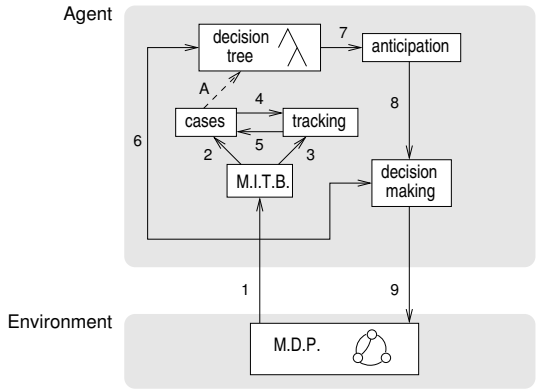


Figure 1: Agent architecture. The relationships among the several modules are: (A) decision tree generation or refinement; (1) stimuli the agent is exposed to; (2) storing a case, after an unexpected negative DV; (3) tracking the differences of a recalled case; (4) recalling a case; (5) storing a tracked case; (6) consulting the decision tree; (7) anticipating future consequences of actions; (8) using anticipations to choose a course of action; (9) action.

Preliminary results

To test this simple agent, a synthetic environment was constructed, using a Markov Decision Process (MDP) as a symbol generator. The agent runs through three periods, in online mode, intercalated by two offline periods in between. These three online periods have all the same number of stimuli (a parameter of the experiment). The MDP is not reset between the online periods, and no symbol is generated during the offline periods. The idea behind this scheme is to provide a first online period where the agent is able to collect cases, a second period to test the causal model generated, where the agent performs an (built-in) `AVOID` action whenever it anticipates a negative DV stimulus, and a third period where it can choose courses of action based on the action consequences collected during the second period.

The MDP used to obtain the results presented in this paper can be found in figure 2. To correctly anticipate the `X` symbol, in this Markov chain, the agent has just to look for a `B` symbol, followed by any symbol (irrelevant), followed by a `D`, and followed by another irrelevant symbol. Whenever this happens, an `X` symbol follows immediately with probability equal to one, unless an `AVOID` action is performed. Since the causal model used by the agent does not account for uncertainty, the MDP used in the following experiments was crafted such that there exists a decision tree capable of correctly anticipating the `X` symbol (negative DV).

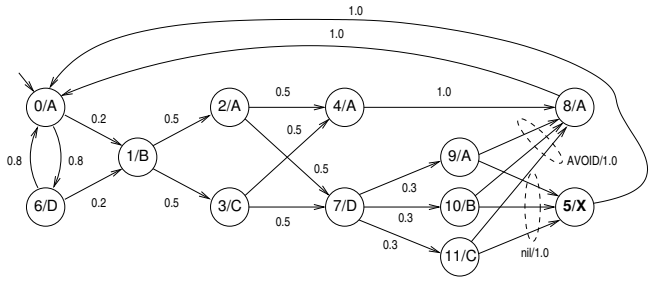
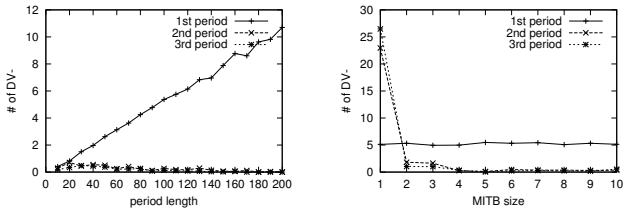


Figure 2: Markov Decision Process (MDP) used to generate a synthetic environment for the agent. The notation used is `state/symbol` inside the state circles, and either the transition probabilities of the corresponding arrows (when the action performed is irrelevant), or the corresponding action/probability pair (when the probabilities depend on the performed action). The state that outputs a `X` is highlighted in bold. The transitions grouped with the dashed ellipsis denote transitions sharing the same action/probability property.

The criterion used to evaluate the agent performance is the number of negative DV symbols the agent was exposed to, during each experiment period. The results presented in figure 3 are plots of this number as a function of the number of stimuli that each online period takes, and of the size[2] of the MITB. In the first plot the MITB size was kept equal to 5, while in the second, the period length was set to 100. The results are presented as averages after running each experiment 100 times.



(a) (b)

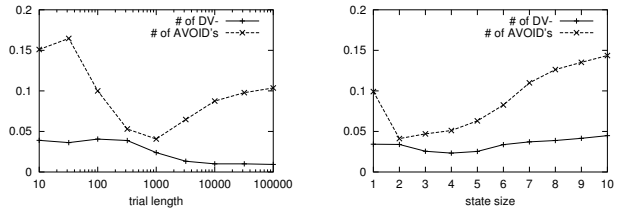
Figure 3: Sensitivity of the agent performance to the length of the online periods (a), and to the size of the "movie-in-the-brain" (b).

Both plots show a performance increase when either the period length, or the MITB size, are increased. During the first period, the number of negative DV stimuli results solely from the MDP statistics, since no DV anticipation is performed because there is no causal model yet. Increasing the period length increases the number of negative DV occurrences. Thus, the agent is able to collect a broader variety

[2]The number of stimuli taken into account to create a case.

of cases in order to build a more informed causal model. Concerning the sensitivity to the MITB size, the number of negative DV stimuli decreases down to around zero as soon as the size is at least 4. This is a consequence of the way the MDP (figure 2) was crafted: a four-step history memory is the minimum required to predict the occurrence of the X symbol. With a sizes of 2 or 3, the D symbol can help anticipating the X, but it can also occur during state (6). However, with a size of 4, the sub-sequence [B, (any symbol), B, (any symbol)] allows a correct anticipation.

Figure 4 shows some results obtained from using a classic Q-learning algorithm (Sutton & Barto 1998) with the same MDP (figure 2) used in this paper as environment. In order to do so, the Q-values were implemented by a table indexed by the string of the latest N symbols concatenated, where N is a parameter. Moreover, the experiments were conducted for two periods: during the first period (exploration), AVOID actions were randomly performed with probability of $1/2$, and during the second one (exploitation), the action performed corresponded to the maximisation of the Q-values. The reward is -1 for the X symbol, -0.1 whenever the agent performs an AVOID action[3], and zero otherwise. The plots in figure 4 show the sensitivity of the number of negative DV stimuli (same performance criterion as before) to the number of stimuli in each period, and to the N is a parameter mentioned above. The results are presented as averages after running each experiment 100 times.



(a) (b)

Figure 4: Sensitivity of the reinforcement learning agent performance to the trial length (a), and to the number of stimuli contained in the state representation (b). The agent performance is measured as the ratio between the amounts indicated and the total number of stimuli in each period.

On the one hand, it is interesting to note how performance degrades as the state dimension (*i.e.,* number of past stimuli contained in the state representation) increases. This effect is usually called "curse of dimensionality." On the other hand, the length of the exploration period required for performance convergence is relatively high: about 3000. Note that one has to be cautious when directly comparing these results with the former ones, since the proposed architecture makes use of built-in knowledge about what to do whenever a negative DV stimulus is anticipated.

Future work

Ongoing work focuses on sophisticating the decision-making mechanism, aiming at, on the one hand, trying out alternative courses of action (in order to circumvent non-trivial environments), and on the other, learning the effects its actions cause in the environment (*i.e.,* "playing"). Playing and exploratory behaviours seem to be useful in the formulation and validation of cause-effect relationships.

[3]This penalisation is needed to prevent the agent from performing AVOID actions all the time.

References

Botelho, L., and Coelho, H. 2001. Machinery for artificial emotions. *Cybernetics and Systems* 32(5):465–506.

Damásio, A. R. 1994. *Descartes' Error: Emotion, Reason and the Human Brain*. Picador.

Frijda, N. H. 1986. *The Emotions*. Paris: Cambridge University Press, Editions de la Maison des Sciences de l'Homme.

Gadanho, S. C., and Hallam, J. 1998. Exploring the role of emotions in autonomous robot learning. In Cañamero, D., ed., *Emotional and Intelligent: The Tangled Knot of Cognition*, 84–89.

Minsky, M. 1988. *The Society of Mind*. Touchstone.

Picard, R. W. 1995. Affective computing. Technical Report 321, M.I.T. Media Laboratory; Perceptual Computing Section.

Quinlan, J. R. 1993. *C4.5: Programs for Machine Learning*. San Mateo: Morgan Kaufmann.

Sloman, A. 1999. Beyond shallow models of emotion. In *i3 Spring Days Workshop on Behavior planning for life-like characters and avatars*.

Staller, A., and Petta, P. 1998. Towards a tractable appraisal-based architecture. In Cañamero, D.; Numaoka, C.; and Petta, P., eds., *Workshop: Grounding Emotions in Adaptive Systems*, 56–61. SAB'98: From Animals to Animats.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement Learning*. MIT Press.

Velásquez, J. D. 1999. From affect programs to higher cognitive emotions: An emotion-based control approach. In Velásquez, J., ed., *Workshop on Emotion-Based Agent Architectures (EBAA'99)*, 114–120.

Ventura, R., and Pinto-Ferreira, C. 1998. Emotion-based agents. In *Proceedings AAAI-98*, 1204. AAAI.

Ventura, R., and Pinto-Ferreira, C. 2002. From reactive to emotion-based agents: a prescriptive model. In Trappl, R., ed., *Cybernetics and Systems 2002*, 757–761. Austrian Society for Cybernetic Studies. Proceedings of EMCSR-2002, Vienna, Austria.

Ventura, R.; Custódio, L.; and Pinto-Ferreira, C. 1998. Emotions — the missing link? In Cañamero, D., ed., *Emotional and Intelligent: The Tangled Knot of Cognition*, 170–175.

Ventura, R.; Custódio, L.; and Pinto-Ferreira, C. 2001. Learning courses of action using the "movie-in-the-brain" paradigm. In *Emotional and Intelligent II: The Tangled Knot of Social Cognition*, 2001 AAAI Fall Symposium, 147–152. AAAI.