Action selection for stochastic, delayed reward*

Herbert Jaeger

GMD – German National Research Center for Information Technology

Abstract. The paper gives a novel account of quick decision making for maximising delayed reward in a stochastic world. The approach rests on *observable operator models* of stochastic systems, which generalize hidden Markov models. A particular kind of decision situations is outlined, and an algorithm is presented which allows to estimate the probability of future reward with a computational cost of only $O(im)$, where i is the number of action alternatives and m is the model dimension.

1 Introduction

Humans can roughly assess the future benefit of an action very quickly and in novel situations. I give two examples. (1) An expert chess player who sees a novel chess configuration can judge "at a glance" the potential benefit of a particular move. (2) A football player during a developing attack runs toward a position where he feels he might be needed some ten seconds later.

These examples have several traits in common:

1. Reward of current action is delayed by an unknown amount of time.
2. The consequences of action are only stochastically predictable. The subject's performance capabilities result from long-time experience and appear to be linked to some (probabilistic) learning capability.
3. The current situation is a novel constellation which was not encountered in prior experience.
4. Decisions are taken very quickly.

Many attempts have been made to model decision-making for achieving a goal or reward, among them the following.

– In cognitive science and artificial intelligence, looking ahead for maximizing reward is often framed as a symbolic, rational planning problem. The planning paradigm is well-suited to model points 1 and 3. However, the current state of the art does not permit to account for both points 2 and 4: either one has stochastic world models with slow (Bayesian) inference, or deterministic world models with fast inference (Kautz & Selman, 1996).

* accepted for KogWis-99, Bielefeld, September 1999

– In machine learning and robotics, determining reward-optimal decisions is often achieved by using reinforcement learning in general and Q-learning in particular (Sutton, 1991). In a learning phase, a random exploration of action-situation sequences enables the agent to build a probabilistic finite state machine model of the world. In the application phase, this models allows to take immediate decisions. A serious shortcoming is bad scaling of learning time with size of world models. Reinforcement learning accounts for points 1, 2, 4 but not for 3.
– In biocybernetics and neurobiology, several neural network architectures have been proposed to model the learning of action sequences with delayed reward (Corbacho & Arbib, 1995) (Klopf, Morgan, & Weaver, 1993). Typically these neural networks account for classical or operand conditioning of short, highly stereotyped action sequences. Points 1, 2, 4 are elegantly explained, but 3 leads out of the realm of stereotyped behavior.

In this article I make a new attempt to answer the old question of how optimising delayed reward. The new approach grows from a recently discovered mathematical model of stochastic systems, named "observable operator models" (OOMs) (Jaeger, 1999). From an information theoretic perspective, OOMs describe stochastic time series prediction as the evolution of a knowledge state which in a strict sense *is* the agent's expectation about the future, as learnt from the past. All of the above four points are accounted for (of course that is why I listed just those).

The article is structured as follows. First I further clarify the scope of phenomena that the paper addresses (section 2). After a sketch of OOM theory (section 3), the main result in section 4 describes how one can quickly compute an estimate of the future reward brought about by a present action.

2 What kind of phenomena, exactly, is addressed

A single, universal cognitive model of how humans assess the consequences of their actions can hardly be expected to exist. The four points given in the introduction circumscribe only roughly the particular kind of future assessment envisaged in this article. In this section, I will highlight two characteristic properties of the particular kind of assessment processes that the OOM approach aims to model.

Self-image. Human experts not only know how situations typically change through their actions. They also know how they themselves will typically act in given situations. They have a self-image about how they respond to situations. This is heavily exploited by the OOM model. Therefore, it cannot capture assessment processes in novices, who have yet to learn how they will typically act.

Forward vs. recurrent decisions. I can discern (at least) two kinds of action decisions, which I would like to call tentatively "forward" and "recurrent" decisions. Forward decisions are the kind of decisions that we take almost

all of the time in the *situated action* (Suchman, 1987) everyday life, for instance when we decide to turn the ignition key when we are seated in the car. Forward decisions occur in non-novel situations and involve no explicit reasoning about their consequences. Recurrent decisions, by contrast, have the property that their expected consequences explicitly feed back into the decision process. They are triggered by novelty in situations. From a mathematical prespective, the outcome of forward decisions is a (stochastic) *function* of the current situation and its prior history. Recurrent decisions, then, should be mathematically featured as *dynamical systems* with a state; their outcome depends on a decision-internal, autonomous process which circles around models of the predicted futures implied by current decision (extensive theoretical treatment in (Rosen, 1985), connectionist model in (Townsend & Busemeyer, 1995)). Arguably, human experts operate in forward decision mode most of the time. The OOM model of future assessment models a single recurrent decision, assuming that in the relevant future only forward decisions will occur.

Thus, in sum, this article aims at scenarios where long situated action sequences of forward decisions are separated by rare, singular recurrent decisions; and where the subject's inclinations for particular forward decisions are part of his/her self-image. The OOM model describes how a single recurrent decision can be made such that maximal reward can be expected during some extended lapse of future, but not later than the next recurrent decision.

3 Background: observable operator models

In this section, I sketch some essentials of OOMs. A detailed treatment is given in (Jaeger, 1999).

OOMs are mathematical models of stochastic processes. The most elementary OOMs, which suffice for the present purpose, describe processes that unfold in time as a stochastic sequence of finitely many *observable events* $\mathcal{O} = \{a_1, \ldots, a_n\}$. Thus, an empirically observed (or mentally envisaged) system history would be a sequence $S = a_{i_1} \ldots a_{i_N}$, where $a_{i_j} \in \mathcal{O}$. A *k-sequence* is a sequence of length k. $\mathcal{O}^k$ denotes the set of all k-sequences. A nonempty subset $A \subset \mathcal{O}^k$ is called a *k-event*. For example, taking $\mathcal{O} = \{a, b, c\}$, the set $\{aa, ab, ac\}$ is a 2-event.

An OOM has three ingredients: (i) a *state space* $\mathbb{R}^m$, (ii) a family $(\tau_a)_{a \in \mathcal{O}}$ of linear operators $\tau_a : \mathbb{R}^m \to \mathbb{R}^m$, and (iii) a *starting vector* $w_0 \in \mathbb{R}^m$. The operators τ_a must obey certain regularity conditions. Note that for every observable event $a \in \mathcal{O}$ there exists an operator τ_a.

The m axes of an OOM state space can be labeled by k-events which partition $\mathcal{O}^k$. For instance, a two-dimensional OOM with $\mathcal{O} = \{a, b, c\}$ could be labeled by the 2-events $A_1 = \{aa, ab, ac, ba, ca\}, A_2 = \{bb, bc, cc\}$. The events $A_1, \ldots, A_m$ pertaining to the labelling of the m dimensions are called the *characteristic events* of the OOM. We use the notation $\mathcal{A}(A_1, \ldots, A_m) = (\mathbb{R}^m, (\tau_a)_{a \in \mathcal{O}}, w_0)$

to denote an m-dimensional OOM with characteristic events $A_1, \ldots, A_n$ and observables $\mathcal{O}$.

The observable operators, being linear maps on $\mathbb{R}^m$, can be represented by $m \times m$ matrices. I will not distinguish between matrices and formal operators.

I use an example to demonstrate how an OOM can be used to compute probabilities of future observations. Assume that one wishes to compute the probability that the sequence aab will occur. We use the shorthand $P[aab]$ to denote this probability. The central law of OOMs is exemplified in the following equation:

$$P[aab] = \mathbf{1} \circ \tau_b \circ \tau_a \circ \tau_a \circ w_0, \tag{1}$$

where $\circ$ denotes matrix-matrix or matrix-vector multiplication, and $\mathbf{1}$ is the $m-$unit-row vector $(1, \ldots, 1)$.

Alternatively, $P[aab]$ can be obtained by first computing $\tau_{aab} = \tau_b \circ \tau_a \circ \tau_a$, and then obtaining $P[aab] = \mathbf{1} \circ \tau_{aab} \circ w_0$. If one wishes to compute the probability that the 3-event $A = \{aab, aba\}$ will occur, one can similarly put $\tau_A = \tau_{aab} + \tau_{aba}$ and obtain $P[A] = \mathbf{1} \circ \tau_A \circ w_0$. "Compiling" sequences or k-events into single operators in this fashion is useful if the probability of this sequence or k-event has to be estimated repeatedly.

Two OOMs are *equivalent* when they yield the same probabilities for event sequences. For any given OOM there are (very) many equivalent ones. Given two partitions $\mathcal{O}^k = A_1 \cup \cdots \cup A_m = B_1 \cup \cdots \cup B_m$, the OOM $\mathcal{A}(A_1, \ldots, A_m)$ can be transformed into an equivalent OOM $\mathcal{A}(B_1, \ldots, B_m)$ essentially with the cost of computing the "compiled" operators τ_{B_i}.

The example in equation (1) brings out clearly how sequences of operators correspond to sequences of observable events in OOM theory, and why the mathematical model is called "observable operator model".

OOMs are a generalization of the widely used hidden Markov models (HMMs). Every HMM can be directly transformed into an equivalent OOM. Thus, the results reported in the next section straightforwardly apply also in cases where the learnt world model is a HMM, (or, equivalently, a probabilistic finite-state machine).

Of course, the probabilities $P[a_{i_1} \ldots a_{i_N}]$ that one obtains from a particular OOM depend on the particular matrices τ_a and the starting vector w_0. In order to model an observed empirical system with an OOM, these matrices and the starting vector have to be estimated (= learnt) from empirical observations in the first place. A very efficient, asymptotically correct learning algorithm is available for OOMs: given a teaching sequence $a_{i_1} \ldots a_{i_N}$, an m-dimensional OOM can be estimated with time complexity $O(N + nm^3)$. The algorithm is described in (Jaeger, 1999).

In equation (1), the starting state w_0 is successively transformed into $\tilde{w}_1 = \tau_a \circ w_0, \tilde{w}_2 = \tau_a \circ \tilde{w}_1, \tilde{w}_3 = \tau_b \circ \tilde{w}_2$. We put $w_1 = P[a]^{-1}\tilde{w}_1, w_2 = P[aa]^{-1}\tilde{w}_2, w_3 = P[aab]^{-1}\tilde{w}_3$. The sequence w_1, w_2, w_3 is called a sequence of *state vectors* of the OOM, which corresponds to the sequence of observations a, a, b.

State vectors of an OOM can be interpreted as knowledge states. Imagine that an agent (human or robot) has learnt from experience an OOM $\mathcal{A}(A_1, \ldots, A_m)$ for some domain with observable events $\mathcal{O}$. Assume that after learning, the OOM is used to monitor an ongoing process for some time – for instance, an initial sequence aab is empirically observed, and the OOM is taken through the state vectors w_0, w_1, w_2, w_3 concurrently, and is finally in state w_3. Then, the agent can use the current state w_3 to compute probabilities of what is going to be observed next. For instance, $P[c \,|\, aab] = \mathbf{1} \circ \tau_c \circ w_3$ is the conditional probability that c is going to happen next, given that aab has happend so far. Likewise, the agent can use w_3 to compute the conditional probability of any other future event. Thus, w_3 comprises all the prediction-relevant knowledge the agent has, at this moment, about the world state.

4 Algorithm: fast computation of optimal action

The state vector $w_3 = (w_3^1, \ldots, w_3^m)$ obtained at time $t = 3$ has an important additional property. Namely, its i-th component is the probability that the i-th characteristic event will be observed in the next k time steps: $w_3^i = P[A_i \,|\, aab]$. This leads to an algorithm for selecting a reward-optimal action. I will explain it along the lines of the running example.

Assume again that the agent has learnt an OOM $\mathcal{A}(A_1, \ldots, A_m)$. This OOM contains the agent's statistical knowledge about the unfolding of situated action sequences, i.e. of forward decision driven agent-environment interaction histories. Assume that the occurence of c is a rewarding event. Assume further that the agent at time $t = 3$ (after the initial world development aab) wishes to make a recurrent decision, and that the agent has the option to make either a or b happen, and that the agent wishes to maximise the chances of c to happen within the next l time steps. Assume furthermore that after $t = 3$, the agent will not further recurrently interfere with the going of things (i.e., will make only forward decisions).

Let $C = \{x_1 \ldots x_l \,|\, x_1 \ldots x_l$ contains at least one $c\}$ be the l-event "reward c occurs". Then, this is an algorithm to decide between a vs. b:

1. Compute τ_C. Due to the particular form of C, this incurs essentially $4l$ matrix multiplications.
2. Transform the OOM $\mathcal{A}(A_1, \ldots, A_m)$ into a version $\mathcal{A}'(C, A'_2, \ldots, A'_m) = (\mathbb{R}^m, (\tau'_a)_{a \in \mathcal{O}}, w'_0)$ with characteristic l-events, the first of which is C, and the others are arbitrary. This essentially incurs m further matrix multiplications.
3. Transform τ_C from the original OOM $\mathcal{A}$ to the equivalent version $\mathcal{A}'$ (one more matrix multiplication).
4. Transform the state vector w_3 into its equivalent state w'_3 in $\mathcal{A}'$ (one vector-matrix multiplication).
5. Compute the first component α of the vector $\mathbf{1} \circ \tau'_a \circ w'_3$, and the first component β of the vector $\mathbf{1} \circ \tau'_b \circ w'_3$. This can be done by $4m$ float operations. It holds that $\alpha = P[C \mid aaba], \beta = P[C \mid aabb]$, i.e. α, β are the estimated

probabilities that a reward event c will occur in the next l time steps after either a or b has been effected by the agent.

The main costs of this procedure stem from the computation of τ_C and the transformation from $\mathcal{A}$ to $\mathcal{A}'$. These "compilation" computations have to be done only once if if the reward event c and the reward "horizon" l remain fixed. Then, only the last step has to be re-computed at each upcoming recurrent decision point – with only $2im$ float operations.

5 Summary

This article proposes an algorithm for fast decision making which optimizes reward in a stochastic world. It rests on a novel representation of stochastic systems, OOMs, which can be efficiently learned from experiential data. The algorithm models a particular (but apparently widespread) kind of decision situations, namely, when a single complex, novel kind of decision is to be followed by a sequence of decisions of a kind which the agent has taken before, such that experiential knowledge about this kind of decisions is available. The algorithm accounts for a combination of factors which was unaddressable so far: (i) delayed reward, (ii) stochastic consequences of action, (iii) novelty of current situation, (iv) quickness of decisions, i.e. low computational complexity.

Acknowledgments. Many thanks to Thomas Christaller for great working conditions, and to Joachim Hertzberg for illuminations on planning. The work described in this article was achieved under a postdoctoral grant from GMD.

References

Corbacho, F., & Arbib, M. (1995). Learning to detour. *Adaptive Behavior*, *3*(4), 419-468.

Jaeger, H. (1999). Observable operator models for discrete stochastic time series. *Neural Computation, to appear.* (Download from http://www.gmd.de/People/Herbert.Jaeger/)

Kautz, H., & Selman, B. (1996). Pushing the envelope: Planning, propositional logic, and stochastic search. In *Proceedings of the 13th National Conference of the American Association for Artificial Intelligence (AAAI-96)* (p. 1194-1201). MIT Press.

Klopf, A., Morgan, J., & Weaver, S. (1993). A hierarchical network of control systems that learn: Modeling nervous system function during classical and instrumental conditioning. *Adaptive Behavior*, *1*(3), 263-319.

Rosen, R. (1985). *Anticipatory systems.* Pergamon Press.

Suchman, L. (1987). *Plans and situated actions.* Cambridge University Press.

Sutton, R. (1991). Reinforcement learning architectures for animats. In J. Meyer & S. Wilson (Eds.), *From Animals to Animats. Proc. 1rst Int. Conf. on the Simulation of Adaptive Behavior* (p. 288-296). Bradford Books/MIT Press.

Townsend, J., & Busemeyer, J. (1995). Dynamic representation of decision-making. In R. Port & T. van Gelder (Eds.), *Mind as motion: Explorations in the dynamics of cognition* (p. 101-120). MIT Press/Bradford Books.