LEARNING OF SENSORY-MOTOR SCHEMAS IN A MOBILE ROBOT

Alan H. Bond and David H. Mott

Artificial Intelligence Laboratory, Queen Mary College,
University of London, Mile End Road, London, El 4NS, England

ABSTRACT

A learning system is described which was used to control a simple robot vehicle and to autonomously learn behaviour patterns. The system is loosely based on Becker's model of Intermediate Cognition.

I INTRODUCTION

There are a number of reasons to regard sensory-motor learning as a basis for intelligence, and therefore to be of greater interest than other forms of learning. Our learning system took Becker's Model of Intermediate Cognition as its starting point and developed and modified it considerably: Becker [l] himself did not implement his model. Mott [2] has criticised Becker's original model. Further details of this work can be found in [3] and [4]. We used a simple but real robot vehicle, the Queen Mary College Mark IV Experimental Robot. It had a set of sensors including touch bars on all four sides, photocells pointing in front of the vehicle and battery level sensors. The robot had a number of other sensors that were not used in this work. The program was written in Pop-2 and ran on an ICL 1904S mainframe computer.

II THE SCHEMA MODEL

Kernels

The system accepted a stream of atomic sensory "kernels"

<BRIGHT>S	from photocells
<FRONT>S	from the front touch bar
<FULL>S	battery fully charged
<HUNGER>S	battery low
<LOW>S	battery very low or overcharged (<HUNGER>S also produced in this case)
<CHARGE>S	battery on charge
<HIGH>S	used internally but not produced by sensing any physical situation

This research was supported by a two-year grant from the United Kingdom Science Research Council.

It generated a stream of motor kernels

<FORW>M	few inches forward
<BACK>M	" backward
<LEFT>M	10 degrees rotation left, about centre
<RIGHT>M	" right "
<CRY>M	bleep Morse code for SOS

B. Short term memory and events

The stream of sensory and motor kernels flowed into an infinitely long queue or STM. The system operated on a discrete time scale, strictly alternating between motor output and sensory input. An item in the queue was the set of kernels occurring at one time, which were thus either all sensory or all motor kernels. A set of kernels occurring at a given time slot was called an event e.g. E={<CHARGE>S <FRONT>S>. An event could be empty. The contents of STM were an event sequence e.g. E_1-->E_2-->-->E_3 where temporal adjacency is denoted by "-►".

C. Schemas and the model

The central data structure was the system's model of the world, which was an unordered set of "schema". A schema was of the form

[event sequence ■> event sequence]

where "=>" means "predicts the occurrence of"

If the left hand side of a schema matched the contents of STM, it was said to occur and the model predicted that the right hand side would occur, starting in the next time slot.

The schema were uncertain, with each kernel having a certainty value for its membership of the schema, and each schema having an overall certainty value of giving a correct description of what happens in STM.

III PREDICTION AND BEHAVIOUR

We regarded Prediction and Behaviour as separate processes. Prediction was the bottom up generation of strings of events starting from the contents of STM and using the model. Instead of producing a prediction tree, it was found easier to verify predictions if they were amalgamated into a linear prediction queue. This was strictly timed with time slots. Predictions had certainty values obtained by multiplying the certainty values of items involved in their creation.

Behaviour was goal seeking if a <LOW>S or <HIGH>S was predicted, otherwise it was exploratory. Goal seeking behaviour tried to achieve <HIGH>S in STM and avoid <LOW>S in STM. It did this by setting up a goal tree top down from a predicted <HIGH>S or <LOW>S as top goal. The subgoals were obtained by chaining back using schemas from the model and each subgoal had a strict time of desired occurrence. Goals also had priorities derived from certainty values.

Negated kernels were found to be useful. A negated kernel -<A> was put into STM if a goal <A> failed to occur. The avoidance of <LOW> was expressed as setting a-i<LOW> top goal. The rules for matching negated kernels were straightforward, since negation meant the absence of that kernel.

A goal succeeded if it was a sensory kernel and this sensory kernel occurred in the corresponding time slot. If it was a motor kernel, it succeeded by the system outputting that kernel and causing an external action of the robot.

Exploratory behaviour involved executing any motor kernels involved in predictions.

IV LEARNING

This occurred in three ways

1. Schema creation

If unpredicted kernels occurred in STM, a new schema was created using these kernels as rhs and the kernels from current time -1 and -2 as the lhs:

TIME	ct-2	ct-1	ct
STM	<A>S<B>S	<M>M	<C>S<D>S
NEW SCHEMA	[<A>S<B>S	--><M>M »>	<C>S<D>S]

with initial certainty values of 1.0.

2. Updating certainty values in existing schema

If a schema partially matched STM, the certainty values were updated. This was only done if all but one of the lhs kernels do match and if at least one of the rhs kernels matched and the average uncertainty of any kernels not matching was less than a threshold (0.4).

The certainty values were calculated in an ad hoc manner as follows:

lhs kernels(k): $\dfrac{\text{no of times k absent and prediction failed}}{\text{no of times k absent and prediction failed or succeeded}}$

rhs kernels(k): $\dfrac{\text{no of times prediction of k succeeded}}{\text{no of times k predicted}}$

overall certainty: 4(1hs contrib + rhs contrib)

where lhs contrib * l ïhs(k)x modi (no of times k absent/10)

and rhs contrib = rhs(k)y modi (no of times prediction of rhs succeeded/10)

where modi(x) ■ if x<l then x else 1.

3. Differentiation

If the certainty value of rhs tended to a stable value which was neither 0 nor 1, this indicated an insufficiently specified context, and extra detail could be added from its occurrence in STM.

Thus if [<A>S--><B>M =><OS]
and if
<X>S->-<Y>M-KA>S-*<B>M-*'<C>S occurs
then the schema is differentiated to
[<X>S+<Y>M-KA>S-*<B>M=><C>S]

V Hi EXAMPLE OF LEARNING

An example of learnt behaviour was "learning to find the light and charge the batteries, when hungry".

The system started with only one "innate" schema viz.

[<FRONT>S-+<FORW>M=><HIGH>s]
i.e. it had a tendency to push against objects in contact with and in front of it.

It learnt
[<HUNGER>S-+ *><HUNGER>S]
i.e. hunger tends to persist
[<HUNGER>S--> =><L0W>S]
i.e. hunger tends to (eventually) produce pain.
[<RIGHT>M =><BRIGHT>s]
this is scanning behaviour i.e. if you turn or "scan" right, you sometimes see the light
[<BRIGHT>S-=><FORW>M =><BRIGHT>S]
this is light beam following i.e. if you see the light and move forwards, you often still see the light.
[<CHARGE>S--><FORW>M =>-i<HUNGER>S]
this is discriminating pushing i.e. if you are on charge and push forwards you may eventually alleviate hunger.

In fact the system, also learnt the more complex form
[<CHARGE>S<FORW>S<HUNGER>S<BRIGHT>S-*<FORW>M *> ■n<HUNGER>S]

This learning required about 350 cycles of the system) about 15 other schema were also learned at the same time but did not interfere.

ACKNOWLEDGEMENT

The authors would like to acknowledge the work of CM. Witkowski in the design and construe-tion of the Mark IV Robot.

REFERENCES

[I] Becker, J.D. "An Information - Processing Model of Intermediate-Level Cognition", Technical Report AI-119, Artificial Intelligence Project, Stanford University, Stanford, California, 1970.

[2] Mott, D.H. "A Criticism of Becker's Model", Technical Report 57, Artificial Intelligence Laboratory, Queen Mary College, London, 1976.

[3] Bond, A.H. and Mott, D.H. "Learning In A Mobile Robot", Final Report to Science Research Council, Technical Report 102, Artificial Intelligence Laboratory, Queen Mary College, London, 1978.

[4] Mott, D.H. "Sensory-Motor Learning In A Mobile Robot", Ph.D. Thesis, University of London (forthcoming), 1981.