ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/220815673

Training and Tracking in Robotics.

Conference Paper · January 1985

Source: DBLP

CITATIONS
82

READS
109

3 authors, including:

Richard Sutton
University of Alberta
175 PUBLICATIONS **44,558** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Computer Go View project

Reinforcement Learning Models View project

All content following this page was uploaded by Richard Sutton on 02 April 2015.

The user has requested enhancement of the downloaded file.

TRAINING AND TRACKING IN ROBOTICS*

Oliver G. Selfridge and Richard S. Sutton
GTE Laba, Waltham, MA 02254

Andrew G. Barto
Department of Computer and Information Science
University of Massachusetts, Amherst, MA 01003

ABSTRACT

We explore the use of learning schemes in training and adapting performance on simple coordination tasks. The tasks are 1-D pole balancing. Several programs incorporating learning have already achieved this (1, S, 8]: the problem is to move a cart along a short piece of track to at to keep a pole balanced on its end; the pole is hinged to the cart at its bottom, and the cart is moved either to the left or to the right by a force of constant magnitude. The form of the task considered here, after (3), involves a genuinely difficult credit-assignment problem. We use a learning scheme previously developed and analysed (1, 7] to achieve performance through reinforcement, and extend it to include changing and new requirements. For example, the length or mast of the pole can change, the bias of the force, its strength, and so on; and the system can be tasked to avoid certain regions altogether. In this way we explore the learning system's ability to adapt to changes and to profit from a selected training sequence, both of which are of obvious utility in practical robotics applications.

The results described here were obtained using a computer simulation of the pole-balancing problem. A movie will be shown of the performance of the system under the various requirements and tasks.

I INTRODUCTION

The importance of good training experience it well recognised in pattern classification and inductive inference, where careful choice of rule exemplars and counter-exemplars clearly affects learning progress (e.g., ref. [9]). It is also important for learning in symbolic problem solving as illustrated by the problem generation component of LEX [5]. Here we show that similar pedagogic care can be significant in non-symbolk problem solving of the kind that is important in robotics; namely, problems of learning to control physical dynamical systems.

It is sometimes faster for a system to Warn to solve a different problem from the one assigned and then to adapt that solution once it it learned. And it it usually far fatter for a system to adapt to new requirements by modifying old solutions than for it to start again from scratch. In the case of learning to control a physical system, it may be much easier, for example, to first learn to control a related system with simpler dynamics, and then to continue to learn at that system is deformed, continuously or by a sequence of steps, into the system required. This it noi an approach that lends itself to orthodox control theory due to the difficulty of adjusting to nnforeseen changes in dynamics and task requirements.

A related issue is that the ability of a system to adjust to unforeseen changes in circumstances and requirements it important even if it does not lead to increased learning speed through training. Although an obvious role of learning is to construct a knowledge base through experience in domains where there is little a priori knowledge, another role that receives less attention is to track a moving optimum, either incrementally or non-incrementally, as unforeseen changes take place.

Here we show how both training and tracking can be done with a learning system that was described by Barto, Sutton, and Anderson [1]. Our domain is the classic problem of balancing a pole in one dimension. This problem has the advantage that it has been considered by several researchers [3, 8], and it can be made quite difficult by adopting certain assumptions. Specifically:

> A rigid pole is hinged to a cart, which is free to move within the limits of a 1-D track. The learning system attempts to keep the pole balanced and the cart within iti limits by applying a force of fixed magnitude to the cart, either to the left or to the right.

This task is susceptible to a number of different learning schemes depending on the quality of the evaluative feedback provided to the learning system. Here we consider a version of the task similar to the one studied by Michie and Chambers [3] in which the only evaluative feedback occurs on failure—hitting the stops with the cart, or having the pole fall over. This sparsity of evaluative feedback create! a genuinely difficult credit-assignment problem. Since a failure usually occurs only after a long sequence of individual control decisions, it is difficult to determine which decisions were responsible for it. To solve this problem, the system identifies certain regions as being in some sense "close" to failure, and will try to avoid them as well. The back-propagation method for generating this internal evaluation is like Samuel's method (6), but refined and improved by Sutton [7].

The learning system usually starts with an empty experiential knowledge base, and it takes some time to fill it with enough data for the system to perform well. We first show that to learn a new task—for example, one with very different parameters, like a vastly heavier pole—can be easier when the system adjusts its knowledge base from a successful state than from scratch. While it is obvious that a suitably primed knowledge base ought to facilitate learning compared to "tabula rata" learning, our point is that the initial knowledge can be that acquired from learning to solve an easier task, which may itself have been learned at a result of facing an even easier task. Other results show this explicitly.

We also consider tasks that add a constraint, such as avoiding particular pole positions. In these tasks, the learning system must adapt to the constraint without explicit instruction as to how to do it. Systems capable of discovering "how" to accomplish a goal, specified only in terms of "what", are more powerful than those that require

* This research was supported by the Air Force Office of Scientific Research and the Avionks Laboratory (Air force Wright Aeronautical Laboratories) through contract F33615-83-1078.

more explicit specifications. Such "operationalization* capabilities are ihown by systems like ours that modify behavior without performing adaptive model reference system identification, as is usual in adaptive :ontrol theoretic techniques. That is, what is used here is a "learning by discovery" method that owes more to AI than to conventional adaptive control.

II TECHNICAL APPROACH AND EXPERIMENTAL RESULTS

Although the state space of the cart-pole system is continuous—-hat is, the state variables are continuous—we divide the four of them discretely, as in [3]:

x:	3 ranges [the position of the cart]
θ:	6 ranges [the angular position of the pole]
$\dot{x}$:	3 ranges [$\dot{x} = dx/dt$]
$\dot{\theta}$:	3 ranges,

which yield together 162 regions corresponding to the combinations. The system is always in just one region. The job of the learning system, then, is to assign the proper action to each region, so that the system will act correctly. The learning algorithm is given in the appendix and is discussed in detail in refs. [1] and [7].

The cart-pole system was simulated on a VAX-11/780, using the equations of motion given in [2] and the following initial parameter values:

cart mass:	1 kg
track length:	4.8 m
pole mass:	0.1 kg
pole length:	1 m
pole stops:	$+12°$, $-12°$
applied force:	10 newtons, left or right.

There are two small coefficients of friction, one for the cart and the other for the pole. In the initial state the pole is stationary and upright, and the cart is stationary in the middle of the track. Since there is always a force, the pole will not long stay upright. The time from initial position to failure—the cart or pole hitting a stop—is considered one learning trial. Time was discretised to a fiftieth of a second. The system was judged to have succeeded in balancing the pole when it achieved a trial that continued without failure for 10,000 time steps, corresponding to some 3.3 minutes real time. We used the number of failures before reaching this criteria of success as some measure of the power and efficiency of learning. A run using the standard parameters listed above typically might have 70 trials before a criterion trial is achieved.

In the first set of tasks the system was required to adapt to changes in the parameters of the cart-pole system. One task required adaptation to bias in the force, which changed from +10 and —10 newtons to +12 and -8, and to +8 and —12. This can be considered an (inaccurate) approximation to tilting the track right and left. That problem was solvable: the control surface learned was specific to the direction of the tilt, and a system trained for one tilt did not work well for another. However, after several switches of the direction of tilt, with training to criterion for each, the system was eventually able to balance the pole each way without failure. That is, the system had generalised its solution to satisfy all the problems simultaneously.

The second task required adaptation to an increase in the mass of the pole. The initial training used the standard .1kg pole to the criterion of 10,000 steps without failure. Then the mass of the pole was increased one order of magnitude (to 1kg). This made the problem much harder, and performance dropped accordingly. However, criterial performance was soon regained: initial training took an average of 68 failures; subsequent adaptation to the heavier pole took only 20. On the other hand, without pre-training, learning to balance the heavier pole took an average of 86 failures. It is clearly more efficient to use a previous solution as the starting place for this new task than to start from scratch.

The system finds it harder to learn to handle shorter poles. If we started training with a pole reduced in length and mass to two-thirds of the original Im/.Ikg pole, it took 119 failures on average to reach criterion, compared to 67 with the full-sise pole. When the system was first trained to criterion on the full-sise pole and then switched to the short pole, however, only 6 additional failures were incurred on average after the switch, for a total of 73 failures overall. This result shows not only adaptability to changing requirements, but also improvement in learning rate with "directed* training.

Directed training resulted in a larger advantage in switching to a shorter track. With the track length 2 meters (instead of 4.8m), learning took more than 250 failures on the average. But training first at 4.8m, then at 4m, 3m, and finally at 2m (each to criterion) took merely 64 + 4 + 9 + 5- 82 failures.

The final task required learning to avoid some region of state space; namely, the region in which the pole is near vertical ($\pm 1°$). We added a penalty, equal to 1/10 the penalty for failure, for being within this region. Despite the fact that remaining in this state is the "natural" way to balance the pole, the learning system reduced the proportion of time spent there from 22% to 8%. Increasing the penalty reduced the fraction still further, but then hitting the stops became more attractive relatively, and balancing failures occurred more often.

III ANALYSIS AND DISCUSSION

The training illustrated in these results consists of selecting a sequence of control tasks that ends with the required task and that presents a graded series of difficulties. Since this kind of sequence consists of entire problem-solving tasks, rather than exemplars and counter-exemplars of a pattern class, it is quite different from the usual training sequence in'supervised learning pattern classification or concept formation. The potential utility of this type of training seems clear; these results show that the learning system considered here is in fact able to benefit from it. The learning system treated here is, moreover, a simple one; it does not deal with hierarchical control at all, nor does it take advantage of a specific accessible knowledge base.

Part of the utility of the training in our examples is due to the fact the external evaluative feedback is in the form of a binary signal—failure or not. Such systems are always subject to what has been termed the "mesa phenomenon" [4]. What is needed is a more continuous feedback so that the system can become better and better, rather than merely satisfy some set of binary constraints. One observes this constantly in people performing physical tasks; it would obviously be desirable in robotics as well. One mechanism used by the learning system to provide a more continuous evaluation is the part of the algorithm that constructs an internal evaluation signal. Training provides another way of reducing the effect of the mesa phenomenon by effectively leading the system to better regions of the solution space. Significantly, this can be done by manipulating aspects of the task without using detailed knowledge of what the performance surface is

like.

Another way to ease the problems caused by the mesa phenomenon is to provide our robotics systems not with just one purpose, but with a structure of purposes: the first purpose to be satisfied is to fulfill the constraints and avoid outright failure—that is, to keep the pole upright and to avoid allowing the cart to crash into the ends of the track. Next the system might try to keep the pole upright, by reducing the average magnitude of *6* ; after that, it might try to minimise the angular velocity, so as to keep the pole still. Beyond that, we can imagine a universal purpose of wanting to do all the above with minimum work, just as people's physical efforts become more and more efficient with practice.

Related to the issue of training, and equally important, is the ability of the learning system to track a changing optimum for the control parameters. In robotics, there are practical reasons for this: the dynamics change with time, sises and viscosities change with temperature, bearings and surfaces suffer wear, and so on. Such problems are clearly not feasible to solve analytically, and in any case, the point is to provide systems whose behavior is robust enough to handle unexpected as well as expected situations.

ACKNOWLEDGEMENTS

The authors thank Charles W. Anderson for assistance with the pole-balancing experiments.

APPENDIX I: The Learning Algorithm

The action of applying a positive force at time t is denoted $y_t = 1$; that of a negative force is denoted $y_t = 0$. The probability π_t that $y_t = 1$ is

$$\pi_t = \sigma(w_t \cdot x_t) = \frac{1}{1 + e^{-w_t \cdot x_t}},$$

where x_t is the received feature vector (indicating the current region of the state space) and w_t is a modifiable parameter vector the same length as x_t. The probability that $y_t = 0$ is $1 - \pi_t$. $\sigma(\cdot)$ is the sigmoidal logistic distribution with the properties that $\sigma(0) = .5$, $\sigma(\infty) = 1$, and $\sigma(-\infty) = 0$. Initially $w_0 = 0$, so that $\pi_0 = \sigma(0) = .5$. w_t is updated according to

$$w_{t+1} = w_t + \alpha p_{t+1} e_t,$$

where α is a positive learning-rate parameter, and e_t is a modifiable vector, initially zero, and updated:

$$e_t = \lambda_w e_{t-1} + (1 - \lambda_w)(y_t - 1/2)x_t.$$

λ_w, $0 \leq \lambda_w < 1$, is a trace-decay parameter.

p_t is an internal reinforcement signal generated from the external reinforcement signal, sensory input, and past experiance. It is defined by

$$p_{t+1} = r_{t+1} + \gamma v_t \cdot x_{t+1} - v_t \cdot x_t,$$

where r_{t+1} is the external reinforcement at time step $t+1$, $0 \leq \gamma < 1$ is a scalar discount-rate parameter, and v_t is a vector recursively generated by

$$v_{t+1} = v_t + \beta p_{t+1} \bar{x}_t, \qquad v_0 = 0.$$

β is a second positive learning-rate parameter, and

$$\bar{x}_t = \lambda_v \bar{x}_{t-1} + (1 - \lambda_v)x_t,$$

where $0 \leq \lambda_v < 1$ is a second trace-decay parameter. See [1, 7] for discussion of these equations. All the results reported here were obtained using the values $\alpha = 1000$, $\beta = 0.5$, $\gamma = 0.95$, $\lambda_w = 0.9$, and $\lambda_v = 0.8$.

REFERENCES

[1] Barto, A. G., Sutton, R. S., *k* Anderson, C. W. Neuronlike elements that can solve difficult learning control problems. *IEEE Trans, on Systems, Man, and Cybernetics,* vol. SMC-14, 1984, 834-846.

[2] Cannon, R. H. *Dynamics of Physical Systems.* New York: McGraw-Hill, 1967.

[3] Michie, D., *k* Chambers, R. A. BOXES: An experiment in adaptive control. In Dale, E., *k* Michie, D. (Eds.), *Machine Intelligence S.* Edinburgh: Oliver and Boyd, 1968, 137-152.

[4] Minsky, M. L. *k* Selfridge, O. G. Learning in random nets. In C. Cherry (Ed.), *Information Theory: Fourth London Symposium.* London: Butterworths, 1961.

[5] Mitchell, T. M. Learning and problem solving. *Proc. IJCAI 8S)* 1983, 1139-1151.

[6] Samuel, A. L. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development,* 1959, 3, 210-229.

[7] Sutton, R. S. Temporal aspects of credit assignment in reinforcement learning. Univ. of Massachusetts Technical Report 84-2, 1984.

[8] Widrow, B., *k* Smith, F. W. Pattern-recognising control systems. In Tow, J. T., *k* Wilcox, R. H., *Computer and Information Sciences.* Washington D. C: Spartan Books, 1964, 288-317.

[9] Winston, P. H. Learning structural descriptions from examples. In Winston, P. H. (Ed.) *The Psychology of Computer Vision.* New York: McGraw-Hill, 1975, 157-209.

View publication stats