Why Conditional Approach is Hard to Learn

Chris Thornton
Cognitive and Computing Sciences
University of Sussex
Brighton BN1 9QN
Email: Chris.Thornton@cogs.susx.ac.uk
Tel: (44)27 678856

December 14, 1994

CSRP 359

Abstract

The paper presents the results of an empirical study in which supervised learning algorithms were used to train an animat to perform a difficult navigation task. The results of the study are explained in terms of a theoretical distinction between two learning-complexity classes.

1 Introduction

Recently there has been increasing interest in the use of learning for the automatic acquisition of animat behaviors. Such behaviours are typically 'choicefull', which means that they entail (or allow for) a variety of responses in some or all situations. With choicefull behaviours it is convenient to use reinforcement learning regimes rather than supervised regimes, since without knowing each situation's correct response it is difficult to draw up a training set. With 'choiceless' behaviours, on the other hand, each situation does have a correct response. The derivation of a training set and implementation of a supervised regime is thus possible. In fact, a supervised regime applied to a choiceless behaviour should always perform at least as well as any reinforcement regime. The only difference between the supervised regime and the reinforcement regime (applied to a choiceless behaviour) is the fact that the feedback provided by the reinforcement regime is a *degraded* (e.g., noisy or intermittent) signal of the correctness of a given action/response. Thus for a reinforcement regime to outperform a supervised regime on a choiceless behaviour, it would have to be the case that the acquisition performance improves while the quality of the feedback signal degrades. This is clearly absurd.

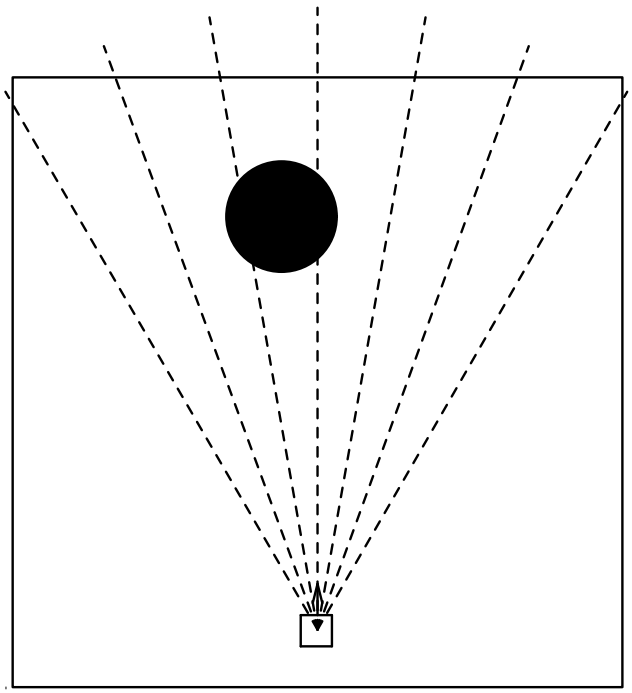
The present paper looks at the learning of a choiceless behavior dubbed 'conditional-approach' by a variety of supervised methods. No reinforcement methods were examined on the assumption that they could not be expected to perform any better than the supervised methods. The behaviour was modeled in an animat with a simple sensory system and a motor system enabling forward and rotational moves. The behavior itself, which involves moving in on any small object in the sensory field but 'standing clear' of any large object, seems rather straightforward. However, it turns out to be poorly learned by supervised methods. We explain this 'failure-to-learn' using a statistical analysis.

The paper is divided into six main sections. This, the first section, is an introduction. In the second section we describe the comparative study, the simulation setup used, the training-data derivation method and the results obtained. In the third section we review basic methods for analyzing statistical properties of training sets and make a distinction between two types of generalization effect. In the fourth section we analyze statistical properties of training sets for the conditional-approach behavior and explain why it is hard to learn. In the fifth section we speculate on the form of general solutions to the conditional-approach learning problem. In the sixth and final section we offer a summary and some concluding comments.

2 he comparative study

The empirical basis of the paper is a comparative survey that investigated a behavior called 'conditional-approach'. The production of this behavior in an animat requires a proximity sensing system of some sort and motor abilities enabling forward and rotational movements. The behavior involves moving in on any relatively small object in the sensory field but standing clear of (i.e., *not* moving in on) any large object.

The behavior was investigated using



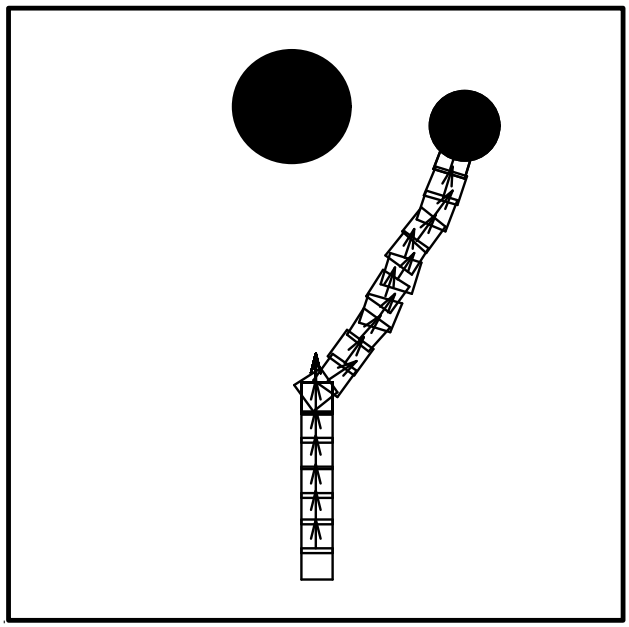


Figure 3: Conditional-approach behaviour.

10% noise. The amount of drive applied to the two wheels in each simulation step was represented in the form of two real numbers, also in the range 0.0-1.0. Thus, a full right turn with no forwards motion would appear in the training set as the pair <1.0,0.0> (given the assumption that the first number sets the drive on the left wheel and the second number the drive on the right wheel).

A sample of training pairs derived for the conditional-approach task is shown in Table 1. Note that the first seven numbers in each row (training pair) are the noisy proximity inputs. These are labeled v1, v2, v3 etc. The final two numbers in each row specify the required amount of drive to be applied to the two drive wheels. These are labeled d1 (amount of drive to the left wheel) and d2 (amount of drive to the right wheel). The first row shows a case of 'standing off' from a large object: the amount of drive for both wheels is 0.00. The second row illustrates the default behavioral response (swivel ten degrees to the right) produced whenever all the proximity inputs are zeros (indicating no object has been sensed). The swivel effect is achieved by setting the amount of drive for the right wheel to be 0.3.

2.4 Algorithms and parameter settings

The use of standard-format training sets enabled us to test the performance of any supervised learning algorithm on the conditional-approach problem. In practice we tested the performance of a wide range of algorithms including ID3 [1] and C4.5 [2], feed-forward network learning algorithms of the backpropagation family including 'vanilla' backpropagation [3], a second-order method based on conjugate-gradient descent [4] and a second-order method based on Newton's method called 'quickprop' [5]. We also tested a constructive network learning method called 'cascade-correlation' [5] and a classifier/genetic-algorithm combination based on Goldberg's 'simple classifier system' [6].

v1	v2	v3	v4	v5	v6	v7	d1	d2
0.00	0.00	0.00	0.27	0.38	0.33	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.3	0.00
0.00	0.00	0.81	0.81	0.81	0.79	0.78	0.00	0.00
0.00	0.87							

2.5 Results

The results can be roughly summarized by saying that C4.5 and nearest-neighbors performed better on the learning task than the connectionist algorithms or the classifier system, but that none of the algorithms provided satisfactory performance on this problem. In general, following training the animat would tend to either approach all objects (large or small) or no objects. It would only very occasionally produce the desired discrimination between large and small objects.

We measured the success of the training in several ways. First of all we measured conventional error rates (i.e. proportion of incorrect responses on unseens). However, these figures give a misleading impression of success. The majority of responses in the conditional-approach behavior do not entail making the crucial discrimination between large and small objects. They merely involve continuing rotatory behavior or moving further towards a small and/or distant object. A better performance measure is provided by sampling the frequencies with which the animat actually arrives at large and small objects. The former frequency we call the 'nip frequency', the latter the 'meal frequency'. These frequencies tend to show the extent to which the animat's behavior embodies the necessary size discrimination.

Our main results are summarized in Table 2. The figures in the 'hand-sim' row show the performance of the animat running under the control of the four rules shown above. The figures in the 'random' row show the performance obtained using a random move generator. The figures in the 'quickprop' row show performance after training with the 'quickprop' version [5] of the backpropagation algorithm [3]; the figures in the row labeled 'c4' show the performance after training with the C4.5 version of ID3 [2]; the figures in the row labeled 'NN' show performance after training with the nearest-neighbours algorithm [7]; finally, the figures in the row labeled CS show performance after training with the simple classifier system/genetic algorithm. All the figures are averaged over 10 different runs. The results reported were gathered using training sets containing 80 training examples since trial and error showed that this size of training set was sufficient to achieve negligably low error on the training examples from all of the algorithms tested.

	Error rate	Meal freq.	Nip freq.
hand-sim		0.864	0.090
random		0.014	0.043
quickprop	0.221	0.201	0.321
c4	0.233	0.479	0.371
NN	0.161	0.117	0.191
CS	0.344	0.251	0.275

Table 2:

The lowest error rate on the testing cases was 0.161 (16.1%) and this was produced by the nearest-neighbours algorithm (NN). This figure seems low but actually reveals relatively poor performance (for reasons

2.6 Comparison with other behaviors (obstacle-avoidance and pursuit)

In view of the possibility that the poor performance obtained from the learning algorithms was due to some flaw in the overall methodology, we carried out some additional experiments. These aimed to discover if we could use the same algorithms and the same methodology to learn more familiar animat behaviors. In particular, we tested the learning algorithms on 'obstacle-avoidance' and 'pursuit'.

For these experiments we used the parameter settings for the learning algorithms described above except in the case of network learning algorithms applied to the obstacle-avoidance task, where we used feed-forward architectures containing just two hidden units with complete connectivity between layers. We also used the same simulation setup but with a modified animat in the case of the obstacle-avoidance training. This animat had the usual two-wheel drive system but it used a simplified sensory system embodying just two proximity probes arranged in a 10 degree, front-facing arc (i.e., it had one probe ray offset five degrees on each side of the forwards direction). The environment for the obstacle-avoidance training was also modified so as to contain three, oval or rectangular objects. The environment was also configured so that the boundaries of the space appeared opaque to the animat. Thus, the simulated animat was able to 'see' both the edges of the objects and the edges of the world.

The control procedure for the obstacle-avoidance simulations was as follows.

(1) Find the higher of the two proximity inputs.

(2) If this value exceeds 0.8 then swivel ten degrees to the right.

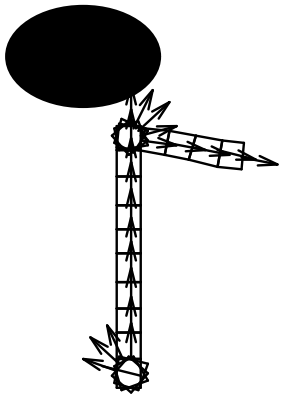
(3) Otherwise move forwards by an amount proportional to the length of the animat.

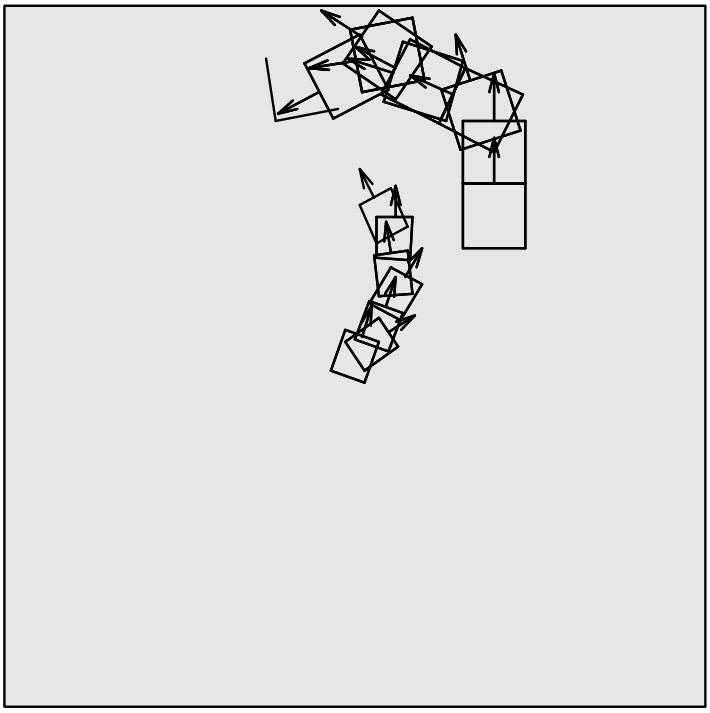
In Figure 4 we see a short trace of a simulated animat producing obstacle-avoidance behavior. The animat's position in each simulation step is shown as a small, arrow-topped box as before. Thus the sequence of boxes shows the animat's trajectory around the environment. Note how the trajectory steers clear of all the obstacles and the boundaries of the space.

2.6.1 Pursuit

The second behavior examined was 'pursuit'. For this behavior we used exactly the same experimental setup as for conditional-approach; i.e., we used a simulated animat with seven probe rays arranged in a 100-degree arc. The environment contained no objects and its boundaries were transparent.

Within the training simulation, the animat tracked a second simulated animat. The shape of this second animat was rectangular and its size was arranged such that it would just intersect two of the seven probe rays at 75% of the maximum animat-to-animat distance. The second animat (henceforth the 'leading animat') moved around the environment according to the following probabilistic regime. In each cycle, there was a probability of 0.3 of the leading animat moving forwards, a probability of 0.35 of it making a forwards+left move and a probability of 0.35 of it making a forwards+right turn. The step size for the leading animat (i.e., the total amount of drive that could be applied to the wheels) was arranged to be 125% that of the pursuing animat. Thus the





3 Learning as the exploitation of justification

The process of learning implemented by some arbitrary learner mechanism can be conceptualized as the acquisition of a target input/output mapping.[2] To have any chance of success the learner requires some source of feedback regarding the mapping to be acquired. In the much studied supervised learning scenario, this feedback takes the form of a set of examples taken from the target mapping. The learner's aim is to arrive at the point at which it is able to map any input taken from the mapping onto its associated output. In more general terms, the learner's aim is to be able to give a high probability to

C	P(C)
	1
x2=2	0.5
x2=1	0.5
y1=1	0.5
y1=0	0.5
x1=3	0.33
x1=2	0.33
x1=1	0.33

Table 4:

C	P(C)
x2=2, y1=1	0.33
x2=1, y1=0	0.33
x1=3, x2=2	0.17
x1=2, x2=2	0.17
x1=3, y1=1	0.17
x1=3, x2=1	0.17
x1=1, x2=2	0.17
x1=2, y1=1	0.17
x1=2, x2=1	0.17
x2=1, y1=1	0.17
x1=1, y1=1	0.17
x1=1, x2=1	0.17

Table 5:

For example, we can observe the conditional probability of observing a particular instantiation of the output variable for given first-order cases of the input variables. These probabilities are shown in Table 6. By the argument used previously, the second-order conditional probabilities here are degenerate since there is necessarily exactly one occurrence of each second-order case of the constrained variables.

In the supervised learning scenario, it is, of course, the conditional and unconditional probabilities re-uandot the

C	P(C)	P(y1=0|C)	P(y1=1|C)
	1	0.5	0.5
x2=2	0.5	0.33	0.67
x2=1	0.5	0.67	0.33
x1=3	0.33	0.5	0.5
x1=2	0.33	0.5	0.5
x1=1	0.33	0.5	0.5

Table 6:

```
Original pairs                Derived pairs (x4 = |x1-x2|)

x1  x2       y1               x4       y1

1   2  -->   1                1  -->   1
2   2  -->   0                0  -->   0
3   2  -->   1                1  -->   1
3   1  -->   0                2  -->   0
2
```

4 Complexity implications for type-1 problems

As I noted above, the aim in supervised learning is to be able to identify target outputs with high probability. We have now seen how the *justifications* for such assignments contained within the learner feedback (i.e., training examples) are either directly or indirectly observed. Discovering direct forms involves deriving probability statistics. Discovering indirect forms involves (1) deriving a recoding of the training examples and (2) deriving probability statistics within the recoded data.

From this we can draw a preliminary conclusion regarding the generic complexity of learning problems. Finding a solution to a particular learning problem necessarily entails discovering some combination of two forms of justification. The number of direct (henceforth 'type-1') justifications is exponentially related to the number of variables and values involved in the training examples.

in the most pronounced conditional probability effects with the output variable(s). ID3's initial aim, therefore, is to identify the input variable which participates in the most informative set of first-order, type-1 justifications. In cases where these justifications provide a satisfactory solution to the problem, the algorithm is guaranteed to find the ideal type-1 solution.

If the type-1 solution involves higher-order probability effects then ID3 is not guaranteed to find the ideal type-1 solution. But in many cases this does not diminish the algorithm's suitability as an approximate type-1 discovery method. Supervised learning problems are typically prepared so as to ensure statistical independence of input variables. If this property is obtained, then higher-order justifications will not exist and any type-1 solution will necessarily be the first-order solution identified by ID3.

Various other learning algorithms exist which provide effective methods for accessing type-1 justifications. Such methods are often biased towards first-order justifications (cf. the Least-Mean-Squares [11] and Perceptron [12] methods) and are thus subject to the same reservations — and recommendations — as the ID3 algorithm. Methods related to backpropagation [3], which do not have such an obvious first-order bias, can potentially be used to find higher-order type-1 solutions (se further discussion below).[5]

5 Complexity implications for type-2 problems

Type-2 justifications are probability effects which are only brought to light via a recoding of the original training data. (Intuitively, they are effects which are discovered via an 'analytic' process.) The fact that there are typically infinitely many possible recodings that might be applied to any given training set means that measuring the difficulty of a type-2 problem (a problem whose solution is based purely on type-2 effects) is impossible unless we set constraints on the nature of the recoding that can be applied.

Before looking at what this means, I will make some preliminary observations about the nature of type-2 effects. Recall that a type-1 effect is an output probability which is either unconditional or conditional on the absolute state of one or more input variables. A type-2 effect is an output probability which is 'associated' in some way with the states of input variables. But it cannot depend in any way on the *absolute* states of input variables since this would make it a type-1 effect! Thus it must be associated with the *relative* states of input variables. Putting it more simply, a type-2 effect must be based on a relational property of the input variables.[6]

This provides us with a rule-of-thumb for deciding whether a learning problem has a type-1 or a type-2 solution. If the problem is based on a relational input/output rule, then probability effects affecting the output variable(s) cannot be based on absolute values of the input variables. Thus the solution can be expected to be comprised of type-2 effects. If the problem is based on a non-relational input/output rule, then the probability effects applying to the output variable must be based on absolute values of the input variables. Thus the solution can be expected to be comprised of type-1 effects.

Of course, in reality, nothing is this simple. Learning problems which are based on relational input/output rules typically have training sets which exhibit 'spurious' type-1 effects. The training set

[5] The Perceptron learning algorithm is, of course, a method which can only be successfully applied to linearly separable problems, i.e., discrimination problems which can be solved by identifying a linear hyperplane separating the relevant classes. Where the hyperplane is aligned with one of the axes of the input space, there will be marked correlations between values of the corresponding input variable and the output variable, and thus pronounced type-1 effects. However, where the hyperplane is unaligned such effects may disappear.

[6] In fact this is not quite true. The argument stated allows it to be based on a non-relational property of a single input variable provided that property effectively masks any type-1 effect.

shown above is a case in point. The input/output rule is relational and yet the type-1 analysis reveals that the training set does exhibit type-1 effects affecting the output variable.

Be this as it may, the basic rule-of-thumb for classifying learning problems does give us a possible explanation for the apparent difficulty of conditional-approach. The input/output rule for conditional-approach is clearly a relational one: it is, after all, based on the calculation of a *ratio* between two values, namely the apparent width and the apparent closeness of the object.[7] The rules for obstacle-avoidance and pursuit, on the other hand, are both based on establishing a direct correspondence between the apparent closeness of an object in the sensory field and a particular behavioral response. In the case of pursuit, the correspondence is a simple matter of sensory stimulation being transformed into drive-wheel activity: objects appearing in particular parts of the sensory field cause particular amounts of drive to be applied to the wheels to ensure an appropriate move/turn. In the case of obstacle-avoidance the correspondence is a matter of sensory stimulation indicative of very near objects inhibiting drive-wheel altogether.

Thus we have a fairly clear distinction between conditional approach on the one hand, and pursuit/obstacle-avoidance on the other. In conditional approach, but *not* in pursuit/obstacle-avoidance, the input/output rule is based on a relational property of the input variables. Any type-1 effects observed in training data for conditional-approach are effectively artifacts, i.e., purely accidental, statistical features attributable to the size/dimensionality/grain etc. of the experimental situation. These artifactual effects may be relatively intense — the complexity of the training set means that measuring them directly is out of the question — but the fact that supervised learners tend to perform badly on this problem, but very well on obstacle-avoidance and pursuit tends to suggest that conditional-approach produces training sets in which the type-1 effects are relatively weak.

6 Summary

The paper presented the results of a comparative study that looked at the conditional-approach behavior. The results of the study showed that several, powerful learning methods are unable to successfully learn the conditional-approach behavior 15999.7(pursfully)]TJ-1612Td(m.4(set)-159.7(pur(successfully)]TJ-18

[3] Rumelhart, D., Hinton, G. and Williams, R. (1986). Learning representations by back-propagating errors. *Nature*, *323* (pp. 533-6).

[4] Becker, S. and Cun, Y. (1988). Improving the convergence of back-propagation learning with second-order methods. CRG-TR-88-5, University of Toronto Connectionist Research Group.

[5] Fahlman, S. and Lebiere, C. (1990). *The Cascade-Correlation Learning Architecture.* CMU-CS-90-100, School of Computer Science, Carnegie-Mellon University, Pittsburgh, PA 15213.

[6] Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley.

[7] Duda, R. and Hart, P. (1973). *Pattern Classification and Scene Analysis.* New York: Wiley.

[8] Nehmzow, U., Smithers, T. and Hallam, J. (1989). Really useful robots. In T. Kanade, F. Green and L. Hertzberger (Eds.), *Proceedings of IAS2, Intelligent Autonomous Systems* (pp. 284-292). Amsterdam.

[9] Cliff, D., Husbands, P. and Harvey, I. (1993). Evolving visually guided robots. In J. Meyer, H. Roitblat and S. Wilson (Eds.), *From Animals to Animats: Proceedings of the Second International Conference on Simulation of Adaptive Behaviour* (SAB92). MIT/Bradford Books.

[10] Millan, J. (forthcoming). On autonomous mobile robots and reinforcement connectionist learning. *Neural Networks and a New AI.* Chapman and Hall.

[11] Hinton, G. (1989). Connectionist learning procedures. *Artificial Intelligence*, *40* (pp. 185-234).

[12] Minsky, M. and Papert, S. (1988). *Perceptrons: An Introduction to Computational Geometry* (expanded edn). Cambridge, Mass.: MIT Press.