An Experimental Comparison of Human and Machine Learning Formalisms

Stephen Muggleton, Michael Bain,
Jean Hayes-Michie, Donald Michie
The Turing Institute, 36 North Hanover St Glasgow G1 2AD, UK.

Abstract

In this paper we describe the results of a set of experiments in which we compared the learning performance of human and machine learning agents. The problem involved the learning of a concept description for deciding on the legality of positions within the chess endgame King and Rook against King. Various amounts of background knowledge were made available to each learning agent. We concluded that the ability to produce high performance in this domain was almost entirely dependent on the ability to express first-order predicate relationships.

1 Introduction

It is a commonly held belief that the use of a restricted hypothesis language simplifies the task of learning. In this paper we investigate a simple problem in which this is not the case. We describe a set of experiments in which a number of different inductive learning agents, with various hypothesis languages, were provided with the same training and test material. In all the experiments described the training and test instances were selected from an instance space of size 262,144 using a standard random generator. Unlike many other comparative studies in the machine learning literature (eg. [2, 7]) we have tested both human and machine learning agents. The machine learning algorithms involved were Quinlan's C4 [6], Clark and Niblett's CN2 [2], Bratko et al's Assistant86 [1], Muggleton's Duce [3] and Muggleton and Buntine's CIGOL [4]. Although CIGOL is capable of constructing new predicates, this feature did not come into play in these experiment. Instead CIGOL's performance excelled in some experiments due to its use of a first-order hypothesis language. In all of these experiments the normally interactive algorithms CIGOL and Duce were run in automatic mode. In this mode oracle questions are automatically taken as being answered positively.

The learning problem involved deciding on the legality of positions in the chess endgame King-and-Rook against King. Despite the fact that predecessors of several of the machine learning algorithms involved were successfully developed and tested using chess endgame problems [5, 3], these same algorithms performed poorly in some of our experiments. The previous successes reported in the literature relied on the provision of special-purpose attributes which encoded relevant features of the board. Typically the use of such attributes dramatically reduces the size of the example space. In one of the experiments described in this paper the only attributes provided were the coordinate values of pieces. This is the lowest level representation of positions used widely by chess players. In this experiment only CIGOL, a first-order learning system, required a small number of training instances to produce reasonable performance on the test set. This shows that the use of a restricted, propositional hypothesis language can prevent concepts from being learned efficiently.

This result confounds a common belief, which could be stated as follows. If a learning algorithm fails to produce predictive performance P within resource-bound R it would also fail to do so with a hypothesis space increased from H to H' ($H' \supseteq H$), since to find a high performance hypothesis you would have to consider at least H. To show the error of this argument consider Figure 1. The tape diagram represents a linear ordering over

Figure 1: Hypothesis ordering, $r = 8$

two hypothesis spaces H and H' where $H' \supset H$. The common ordering represents a simplified version of what is often called learning *bias* [8]. Most practical learning algorithms use a simplicity criterion for this bias. A resource-bounded learning algorithm will only construct the first r hypotheses, limited by its resource bound R. The hypotheses in the diagram are of two different types, which might be thought of as first-order (striped) and propositional (spotted). Whereas a first-order algorithm will construct the r hypotheses $H'(R)$, a propositional algorithm with the same bias would construct the r hypotheses $H(R)$. This contradicts the belief that the first-order algorithm would have to consider at least the hypotheses considered by a propositional learning algorithm since $H'(R) \not\supseteq H(R)$. In the case exemplified by Experiment 1a (section 3.1.1), whereas there is a simple first-order description of part of the concept discoverable within the computational resource bounds, there is no corresponding simple propositional description.

2 Definitions

Terms used in this paper are defined as follows.

Formalism: The syntax of the hypothesis language. **Hypothesis vocabulary:** Predicate symbols used in constructing hypotheses. **Hypothesis language:** The formalism plus the hypothesis vocabulary. **Background knowledge:** Axiomatisation of the individual symbols in the hypothesis vocabulary. **Instance vocabulary** *or* **piece-on-place attributes:** Position of a chess piece described in terms of its file (a-h) or rank (1-8). **Instance language:** An instance is a class value paired with a vector of symbols from the instance vocabulary. **Example language:** An example is a class value paired with a vector of symbols from the instance and/or hypothesis vocabulary. **Hypothesis space:** Set of all possible hypotheses within the hypothesis language. **Instance space:** Set of all possible instances in the instance language. **Example space:** Set of all possible examples in the example language.

3 Experiments

Four experiments were carried out.

1. Learning from piece-on-place attributes.
   (a) Small number of training instances (100). Involved CIGOL, Duce, C4, CN2, Assistant, humans.
   (b) Large number of training instances (1000). Involved Duce, C4, CN2, Assistant.
2. Learning with extended hypothesis vocabulary
   (a) Small number of training examples (100). Involved CIGOL, Duce, C4, Assistant, humans.

Figure 2: Three representations of the illegal position White: Kg6 Rc7; Black: Kc8; White to move.
CIGOL(1a)'s representation is close to standard chess notation. In the representation of Human(1a) positions were coded using two separate substitutions for the "a,b,..,h" and "1,2,..,8" alphabets. The substitutions were respectively: "b,e,h,k,n,q,t,w" and "c,f,i,l,o,r,u,x". This encoding obscures while not removing the ordering of the alphabets. In the Human(2a) grid the symbols for White King, White Rook and Black King are ○, × and ● respectively.

(b) Large number of training examples (1000). Involved Duce, C4, Assistant.

Owing to the diversity of the learning agents involved it was not possible to make the learning conditions identical for each agent. For instance, although it is possible to define and control the background knowledge available to machine learning algorithms, this can only be approximated in humans. Our subjects knew nothing of chess, but inbuilt spatial intuition is bound to have helped their ability to extract relations such as "collinear" from the example data.

3.1 Method

3.1.1 Experiment 1a - Without extended hypothesis vocabulary, small training set

Each machine learning algorithm was trained on five randomly generated sets of 100 instances. Each instance consisted of an illegal/legal class value paired with 6 attribute values. Each attribute value represented respectively the rank and file of each piece. An instance taken from one of the CIGOL training sets is shown in Figure 2.

The five rule sets induced by each machine learning algorithm except CN2 were tested on each of five randomly generated sets of 1000 instances, making 25 tests in total. Each of the five rule sets learned by CN2 was tested on only one set of 1000 examples.

The human subjects consisted of 13 schoolchildren, aged 15-17, 7 boys and 6 girls. These subjects were provided with symbolic descriptions of instances similar to those provided for other learning systems, one of which is shown in Figure 2, Human(1a). However, to avoid errors due to fatigue human subjects were tested on only the first hundred of one of the test sets.

3.1.2 Experiment 1b - Without extended hypothesis vocabulary, large training set

Each machine learning algorithm was trained on five randomly generated sets of 1000 instances. The test regime for all algorithms except CN2 was the same as that of Experiment 1a. Each of the five rule sets learned by CN2 was tested on one set of 100 instances.

Agents	Expt. 1a	Expt. 1b	Expt. 2a	Expt. 2b
Humans	51.2%, 1hr	N/A	79.3%, 1hr	N/A
CIGOL	84.2%, 1.5hr	N/A	77.2%, 21.5hr	N/A
C4	67.0%, 2.5hr	83.4%, 12.2hr	61.9%* 1.6hr	99.0%* 10hr
CN2	69.5%$^+$ 0.4hr	87.6%$^+$ 4hr	N/A	N/A
Assistant	55.7%, 0.25hr	56.2%*, 0.5hr	71.0%*, 0.25hr	91.0%*, 0.5hr
Duce	42.4%, 8hr	47.8%*, 10hr	33.7%*, 2hr	37.7%*, 10hr

Figure 3: Averaged final performance for each agent in each experiment together with approximate mean elapsed time for training and testing.
A '*' appears beside those values for which testing is not complete. A 'N/A' appears in those in which we are not attempting to carry out testing. A '+' indicates a variant training and testing regime, explained below.

3.1.3 Experiment 2a - With extended hypothesis vocabulary, small training set

Each machine learning algorithm was trained on the same randomly generated sets of 100 instances used for training in Experiment 1a, but in some cases extra background knowledge was supplied and in others the examples were presented in an extended hypothesis vocabulary. This was done as follows.

1. CIGOL and Duce - background knowledge predicate definitions for equality, adjacency and less than for files and ranks.
2. C4 and Assistant - hypothesis vocabulary extension consisting of all pairwise arithmetic differences between the integer file and rank values of all pieces.

These algorithms were tested on examples based on the same test instances as those used in Experiment 1a, using the same testing regime.

For the human subjects each example was presented as a diagram of an 8×8 array with circles and crosses in place of the pieces (see Figure 2). Each diagram was marked "yes" (illegal) or "no" (legal). All positions were the same as those appearing in one of the training and test sets of Experiment 1a. At no time was it suggested that the concept being learned concerned chess.

3.1.4 Experiment 2b - With extended hypothesis vocabulary, large training set

Each learning agent was supplied with the background knowledge used in Experiment 2a. The sets of training and test instances were the same as those used in Experiment 1b.

3.2 Results

In order to record the incremental performance change, shown, we tested the performance against the entire test set in increments of 10 training instances. The Experiment 1a incremental performance figures for the all the machine learning algorithms except CN2, averaged over the 25 test runs, are graphed in Figure 4. A summary of the averaged final performance figures for each agent in each experiment together with the approximate mean elapsed time for training and testing is provided in Figure 3.1.1. A breakdown of the performance of individual human subjects together with the significance levels of their performances is provided in Figure 3.2.

4 Discussion

4.1 Experiment 1

4.1.1 CIGOL

The incremental performance graph shown in Figure 4 are in many ways more informative than the final values shown in Figure 3.1.1. In experiment 1a all machine learning

Figure 4: Incremental performance for Experiment 1a

	Humans(1a)	Humans(2a)
Subject performance	71^{***}, 71^{***}, 54, 47, 44, 20^{***}	98^{***}, 96^{***}, 92^{***}, 88^{***}, 64^{*}, 64^{*}, 53
Group mean performance	51.2	79.3

Figure 5: Breakdown of human results.
Subject performances were tested for significance using 2×2 χ^2 test with Yates' correction. A '***' indicates a significance value of $p < 0.001$. A '*' indicates a significance value of $0.05 < p < 0.01$. Unmarked subject performance values are not significant at the 0.05 level.

techniques start from a value of 67%. Since 67% of the instances in the instance space are legal, this is the "null" performance which would be expected from any system which assumes the default "everything is legal". Within the space of around 50 training examples CIGOL's performance rises to an average value of around 85% (91.4% maximum). In doing so CIGOL's hypothesis in Prolog is as follows.

```
illegal(A,B,C,D,A,B).  % The position is illegal iff the White King and the Black King are on the same square or
illegal(A,B,C,D,C,E).  % the White Rook and the Black King are on the same file or
illegal(A,B,C,D,E,D).  % the White Rook and the Black King are on the same rank
```

Within this domain it is possible to analyse how many examples would be necessary for CIGOL to learn any particular unit clause. CIGOL needs at least two examples to be able to form an hypothesis such as "illegal(A,B,C,D,C,E)" by using its *truncation* rule. This would require two instances in which the White Rook and the Black King were on the same file. Imagining that we placed the White Rook on an arbitrary position on a chess board and then placed the Black King on another randomly chosen position the probability that they would lie on the same file is clearly $\frac{1}{8}$. However, CIGOL will often need more than two examples to make the generalisation "illegal(A,B,C,D,C,E)" since there is a $\frac{6}{8}$ chance that any two arbitrarily chosen instances of this rule will have a corresponding rank or file value. This would lead to an hypothesis such as "illegal(A,3,C,D,C,E)", i.e. an under-generalisation. On the basis of this argument we would expect to require between $2 \times 8 = 16$ and $3 \times 8 = 24$ positive examples to develop the rule "illegal(A,B,C,D,C,E)". Since only one in three instances are "illegal" we would expect to require between 48 and 72 randomly chosen examples to develop the two *collinearity* rules "illegal(A,B,C,D,C,E)" and "illegal(A,B,C,D,E,D)". In practice this takes between 40 and 80 examples, much as predicted. The fact that such analysis is possible points to an advantage of carrying out this kind of experimentation within a closed and analytically tractable domain.

It is also easy to see that CIGOL's performance will not ever rise to 100%. The reason is that the collinearity rules, although allowing rapid promotion to high performance, are overgeneralisations. Exceptions exist to these rules when the White King is interposed between the White Rook and Black King. Since CIGOL learns monotonically, it is not

possible to correct such overgeneralisations. Specialisation techniques to overcome this problem are presently under investigation.

4.1.2 CN2 and C4

CN2 and C4 produced very similar performance, with performance almost indistinguishable from the performance of the null rule "every position is legal" in Experiment 1a. Both performances rise gradually when presented with ten times as much training data to a more reasonable 88% and 83% respectively.

4.1.3 Assistant and Duce

The performance of both Assistant and Duce rapidly diminishes from an initial 67%. In Assistant's case performance levels out at 56%, whereas Duce levels out at 48% (Expt 1b).

These poor performances by C4, Assistant and Duce can be partly explained by the fact that a complete description of collinearity within a decision tree propositional formalism is very large and thus needs a large number of examples to justify such an hypothesis. With some work it might be possible to predict just how many examples would be required in the same way as we have done for CIGOL.

4.1.4 Humans

At first sight the human mean performance value given in Figure 3.1.1 looks close to that produced by random guessing. However this is clearly not the case when we look in depth at the individual scores in Figure 3.2. As evidenced by the starred significance indications, individual scores are strongly polarised into those that found an effective prediction method and those which merely guessed. In all cases but one insigificant performances agree with the children's reporting that they merely guessed on the answer sheet. The exception to this is the unexplained 20% score which produces a highly significant score with "YES" and "NO" reversed.

4.2 Experiment 2

4.2.1 CIGOL and Duce

Information on the time taken for each learning agent to reach the performance levels shown in Figure 3.1.1 helps to understand these results. CIGOL typically takes longer to learn by an order of magnitude when supplied with background knowledge than in Experiment 1a. Usually none of the background predicates appear in the final hypothesis so they do not add to predictive power. However the number of predicates in the knowledge base is doubled when the background knowledge is included. This enlarged search space means that CIGOL is unable to find the best hypotheses within available resources.

Duce and CIGOL are machine learning algorithms which construct their own background predicates when doing so simplifies the problem. It was this capability that suggested that they might be appropriate candidates for this learning task. However, we now realise that the achievement of high performance within this domain is *not* dependent on the availability or constructibility of background predicates, but rather a problem of having a sufficiently expressive formalism. This seems to contradict the results of strong performance of C4 and Assistant given appropriate background knowledge. However, a glance at the form of necessary background knowledge for C4 and Assistant's strong performance (Section 3.1.3) shows that it is essentially relational, i.e. could only be expressed in a First-order language. The usually vague notion of "background knowledge" in this case conceals a change of formalism.

4.2.2 Humans

The question of formalism also appears in the human results. When asked to describe the rules that they were applying all successful candidates gave rules similar to the following.

Concept is true
If black nought is in the same line as the cross
If white nought is right next to the black nought
If white and black noughts are in the same box
If black nought is in the same box as the cross

It is clear from this description that relational attributes are used throughout. This supports our main conclusion that the ability to produce high performance in this domain is almost entirely dependent on the ability to express first-order predicate relationships.

Acknowledgements. We would like to thank the staff and pupils of Glenwood School, Glasgow, for their help in carrying out the human learning tests and Pete Clark for carrying out the tests of CN2. We would also like to thank the Turing Institute, Knowledgelink and Attar Software for facilities and help. We also thank the members of the Turing Institute Machine Learning Group for helpful suggestions and discussion. This work was supported partly by the Esprit Ecoles project 3059.

References

[1] B. Cestnik, I. Kononenko, and I. Bratko. Assistant 86: a knowledge-elicitation tool for sophisticated users. In *Progress in machine learning*, pages 31–45, Wilmslow, England, 1987. Sigma.

[2] P. Clark and T. Niblett. The CN2 algorithm. *Machine Learning*, 3(4):261–283, 1989.

[3] S. Muggleton. Duce, an oracle based approach to constructive induction. In *IJCAI-87*, pages 287–292, San Mateo, CA, 1987. Morgan-Kaufmann.

[4] S.H. Muggleton and W. Buntine. Machine invention of first-order predicates by inverting resolution. In *Proceedings of the Fifth International Conference on Machine Learning*, pages 339–352. Kaufmann, 1988.

[5] J.R. Quinlan. Learning efficient classification procedures and their application to chess end games. In R. Michalski, J. Carbonnel, and T. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach*. Tioga, Palo Alto, CA, 1983.

[6] J.R. Quinlan. Generating production rules from decision trees. In *Proceedings of the Tenth International Conference on Artificial Intelligence*, pages 304–307, San Mateo, CA:, 1987. Morgan-Kaufmann.

[7] J.C. Schlimmer and D.H. Fisher. A case study of incremental concept induction. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 496–501, San Mateo, CA, 1986. Morgan-Kaufmann.

[8] P.E. Utgoff. Adjusting bias in concept learning. In *IJCAI-83*, pages 447–449, Los Angeles, CA, 1983. Morgan-Kaufmann.