Analogical Prediction

Stephen Muggleton,
Michael Bain*

Department of Computer Science,
University of York,
YO10 5DD,
United Kingdom.

Abstract. Inductive Logic Programming (ILP) involves constructing an hypothesis H on the basis of background knowledge B and training examples E. An independent test set is used to evaluate the accuracy of H. This paper concerns an alternative approach called Analogical Prediction (AP). AP takes B, E and then for each test example $\langle x, y\rangle$ forms an hypothesis H_x from B, E, x. Evaluation of AP is based on estimating the probability that $H_x(x) = y$ for a randomly chosen $\langle x, y\rangle$. AP has been implemented within CProgol4.4. Experiments in the paper show that on English past tense data AP has significantly higher predictive accuracy on this data than both previously reported results and CProgol in inductive mode. However, on KRK illegal AP does not outperform CProgol in inductive mode. We conjecture that AP has advantages for domains in which a large proportion of the examples must be treated as exceptions with respect to the hypothesis vocabulary. The relationship of AP to analogy and instance-based learning is discussed. Limitations of the given implementation of AP are discussed and improvements suggested.

1 Introduction

1.1 Analogical prediction (AP)

Suppose that you are trying to make taxonomic predictions about animals. You might already have seen various animals and know some of their properties. Now you meet a platypus. You could try and predict whether the platypus was a mammal, fish, reptile or bird by forming analogies between the platypus and other animals for which you already know the classifications. Thus you could reason that a platypus is like other mammals since it suckles its young. In doing so you are making an assumption which could be represented as the following clause.

```
class(A,mammal) :- has_milk(A).
```

* Current address: Department of Artificial Intelligence, School of Computer Science and Engineering, University of New South Wales, Sydney 2052, Australia.

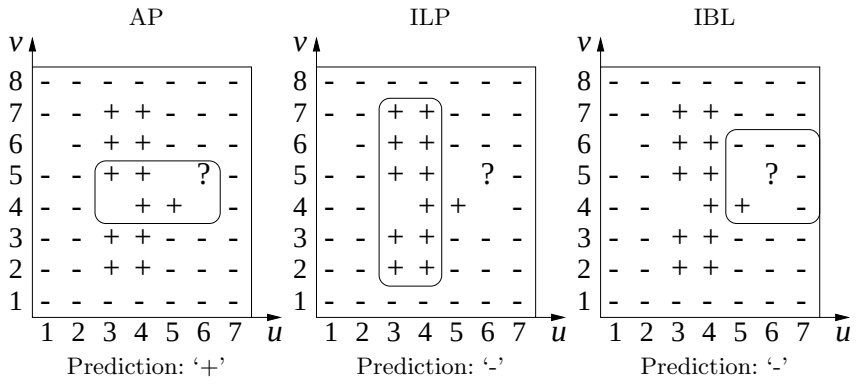

Fig. 1. Comparison of AP, ILP and IBL. Instances x are pairs $\langle u, v \rangle$ from $\{1, .., 7\} \times \{1, .., 8\}$. Each x can have a classification $y \in \{+, -\}$. The test instance x to be classified is denoted by '?'. The rounded box in each case defines the extension of the hypothesis H. Below each box the corresponding prediction for x is given.

It might be difficult to find a consistent assumption similar to the above which allowed a platypus to be predicted as being a fish or a reptile. However, you could reason that a platypus is similar to various birds you have encountered since it is both warm blooded and lays eggs. Again this would be represented as follows.

```
class(A,bird) :- homeothermic(A), has_eggs(A).
```

Note that the hypotheses above are related to a particular test instance, the platypus, for which the class value (mammal, bird, etc.) is to be predicted. We will call this form of reasoning *Analogical Prediction* (AP).

1.2 AP, induction and instance-based learning

In the above AP is given a *test instance* x, a training set E and background knowledge B. It then constructs an hypotheses H_x which not only covers some of the training set but also predicts the class y of x. This can be contrasted with the normal semantics of ILP [10], in which hypotheses H are constructed on the basis of B and E alone. In this case x is presented as part of the test procedure after H has been constructed.

AP is in some ways more similar to Instance-Based Learning (IBL) (see for example [3]), in which the class y would be attributed to x on the basis of its proximity to various elements of E. However, in the case of IBL, instead of constructing H, a similarity measure is used to determine proximity.

Figure 1 illustrates the differences in prediction between AP, standard ILP and IBL on a 2D binary classification problem. The test concept, 'insign', is actually a picture of the symbol '$\vdash$' made out of +'s. If AP is restricted to

making a single clause maximally general hypothesis, it would predicts x to be positive based on the following.

```
insign(U,V) :- 3=<U=<6, 4=<V=<5.
```

Assuming the closed world assumption is used for prediction, normal ILP will predict x to be negative based on the following hypothesis (note the exception at $\langle 5, 4 \rangle$).

```
insign(U,V) :- 3=<U=<4, 2=<V=<7.
```

Finally IBL will predict x to be negative based on the fact that 5/6 of the surrounding instances are negative. Note that IBL's implicit hypothesis in this case could be denoted by the following denial.

```
:- insign(U,V), near(U,V,6,5).
```

The background predicate *near/4* in the above encodes the notion of 'nearness' used in a k-nearest neighbour type algorithm.

1.3 Motivation

AP can be viewed as a half-way house between IBL and ILP. IBL has a number of advantages over ILP. These include ease of updating the knowledge-base and the fact that theory revision is unnecessary after the addition of new examples. AP shares both these advantages with IBL. On the other hand IBL has a number of disadvantages with respect to ILP. Notably, IBL predictions lack explanation, and there is a need to define a metric to describe similarity between instances. Generally, similarity metrics are hard to justify, even when they can be shown to have desirable properties (eg. [5, 11, 12]). In comparison AP predictions are directly associated with an explicit hypothesis, which provides explanation. Also AP does not require a similarity measure since predictions are made on the basis of the hypothesis.

This paper has the following structure. A formal framework for AP is provided in Section 2. Section 3 describes an implementation of AP within CProgol4.4 (ftp://ftp.cs.york.ac.uk/pub/ML_GROUP/progol4.4). This implementation is restricted to the special case of binary classification. Experiments using this implementation of AP are described in Section 4. On the standard English past tense data set [7] AP has higher predictive accuracy than FOIDL, FOIL and CProgol in inductive mode. By contrast, on KRK illegal AP performs slightly worse than CProgol in inductive mode. In the discussion (Section 5) we conjecture that AP has advantages for domains in which a large proportion of the examples must be treated as exceptions with respect to the hypothesis vocabulary. We also compare AP to analogical reasoning. The results are summarised in Section 6, and further improvements in the existing implementation are suggested.

```
Aleave(B,E)
   Let AP=Ap=aP=ap=0
   For each e = ⟨x, y⟩ in E
(a)   Construct ⊥_{B,x}
      E' := E \ e
(b)   Using E' find most compressive H_x ⪰ ⊥_{B,x}
      if y = H_x(x) then
        if y = True then AP := AP + 1
        else ap := ap + 1
      else
        if y = True then Ap := Ap + 1
        else aP := aP + 1
   Print-contingency-table(AP,Ap,aP,ap)
```

Fig. 2. Aleave algorithm. Algorithms from CProgol4.1 are used to (a) construct the bottom clause and (b) search the refinement graph.

2 Definitions

We assume denumerable sets X, Y representing the instance and prediction spaces respectively and a probability distribution $\mathcal{D}$ on X. The target theory is a function $f : X \to Y$. An AP learning algorithm L takes background knowledge B together with a set of training examples $E \subseteq \{\langle x', f(x')\rangle : x' \in X\}$. For any given B, E and test instance $x \in X$ the output of L is an hypothesised function H_x. Error is now defined as follows.

$$error(L, B, E) = Pr_{x \in \mathcal{D}}[h_x(x) \neq f(x)] \tag{1}$$

3 Implementation

AP has been implemented as a built-in predicate *aleave* in CProgol4.4 (available from ftp://ftp.cs.york.ac.uk/pub/ML_GROUP/progol4.4). The algorithm, shown in Figure 3, carries out a leave-one-out procedure which estimates AP error as defined in Equation (1). In terms of Section 2 each left out example e is viewed as a pair $\langle x, y\rangle$ where x is a ground atom and $y =$ True if e is positive and $y =$ False if e is negative.

AP error (see Equation 1) is estimated using the counters AP, Ap, aP and ap ('a' and 'p' stand for actual and predicted, and capitalisation/non-capitalisation stands for the value being True/False). For each example $e = \langle x, y\rangle$ left out, a bottom clause $\bot_{B,x}$ is constructed which predicts $y :=$ True. A refinement graph search of the type described in [8] is carried out to find a maximally compressive single clause H_x which subsumes $\bot_{B,x}$. In doing so compression is computed relative to $E \setminus e$. If no compressive clause is found then the prediction is False. Otherwise it is True.

English past tense	KRK illegality
past([w,o,r,r,y],[w,o,r,r,i,e,d]).	illegal(3,5,6,7,6,2).
past([c,l,u,t,c,h],[c,l,u,t,c,h,e,d]).	illegal(3,6,7,6,7,4).
past([w,h,i,z],[w,h,i,z,z,e,d]).	:- illegal(2,5,5,2,4,1).
past([g,r,i,n,d],[g,r,o,u,n,d]).	:- illegal(5,7,1,2,0,0).

Fig. 3. Form of examples for both domains

English past tense	KRK illegality
past(A,B) :- split(A,C,[r,r,y]), split(B,C,[r,r,i,e,d]).	illegal(A,B,A,B,_,_). illegal(A,B,_,_,C,D) :- adj(A,C), adj(B,D). illegal(A,_,B,_,B,_) :- not A=B. illegal(_,A,B,C,B,D) :- A<C, A<D.

Fig. 4. Form of hypothesised clauses

The procedure Print-contingency-table(AP,Ap,aP,ap) prints a two-by-two table of the 4 values together with the accuracy estimate, standard error of estimation and χ^2 probability.

4 Experiments

The experiments were aimed at determining whether AP could provide increased predictive accuracy over other ILP algorithms. Two standard ILP data sets were chosen for comparison (described in Section 4.2 below).

4.1 Experimental hypotheses

The following null hypotheses were tested in the first and second experiments respectively.

Null hypothesis 1. AP does not have higher predictive accuracy than any other ILP system on any standard data set.

Null hypothesis 2. AP has higher predictive accuracy than any other ILP system on all standard data sets.

Note that hypothesis 1 is not the negation of hypothesis 2. If both are rejected then it means simply that AP is better for some domains but not others.

4.2 Materials

The following data sets were used for testing the experimental hypotheses.

English past tense. This is described in [15, 6, 7]). The available example set E_{past} has size 1390.

KRK illegality. This was originally described in [9]. The total instance space size is $8^6 = 262144$.

For both domains the form of examples are shown in Figure 3 and the form of hypothesised clauses in Figure 4. Note that in the KRK illegality domain negative examples are those preceded by a ':-', while the English past tense domain has no negative examples. The absence of negative examples in the English past tense domain is compensated for by a constraint on hypothesised clauses which Mooney and Califf [7] call output completeness. In the experiments output completeness is enforced in CProgol4.4 by including the following user defined constraint which requires that *past/2* is a function.

```
:- hypothesis(past(X,Y),Body,_), clause(past(X,Z),true), Body, not(Y==Z).
```

4.3 Method

English past tense Mooney and Califf [7] compared the predictive accuracy of FOIL, IFOIL and FOIL on the alphabetic English past tense task. We interpret the description of their training regime as follows. Training sets of sizes 25, 50, 100, 250 and 500 were randomly sampled without replacement from E_{past}, with 10 sets of each size. For each training set E a test set of size 500 was randomly sampled without replacement from $E_{\text{past}} \setminus E$. Each learning system was applied to each training set and the predictive accuracy assessed on the corresponding test set. Results were averaged for each training set size and reported as a learning curve.

For the purposes of comparison we followed the above training regime for CProgol4.4 in inductive mode. We also ran AP with the *aleave* predicate built-in to CProgol4.4 (Section 3) on each of the training sets and then for each training set size averaged the results over the 10 sets.

KRK illegality Predictive accuracies were compared for CProgol4.4 using AP with leave-one-out (*aleave*) against induction with leave-one-out (*leave*). Training sets of sizes 5, 10, 20, 30, 40, 60, 80, 100, 200 were randomly sampled with replacement from the total example space, with 10 sets of each size. For each of the training sets both *aleave* and *leave* were run. The resulting predictive accuracies were averaged for each training set size.

4.4 Results

English past tense The five learning curves are shown in Figure 5. The horizontal line labelled "Default rules" represents the following simple Prolog program which adds a 'd' to verbs ending in 'e' and otherwise adds 'ed'.

```
past(A,B) :- split(A,B,[e]), split(B,A,[d]), !.
past(A,B) :- split(B,A,[e,d]).
```

The differences between AP and all other systems are significant at the 0.0001 level with 250 and 500 examples. Thus null hypothesis 1 is clearly rejected.

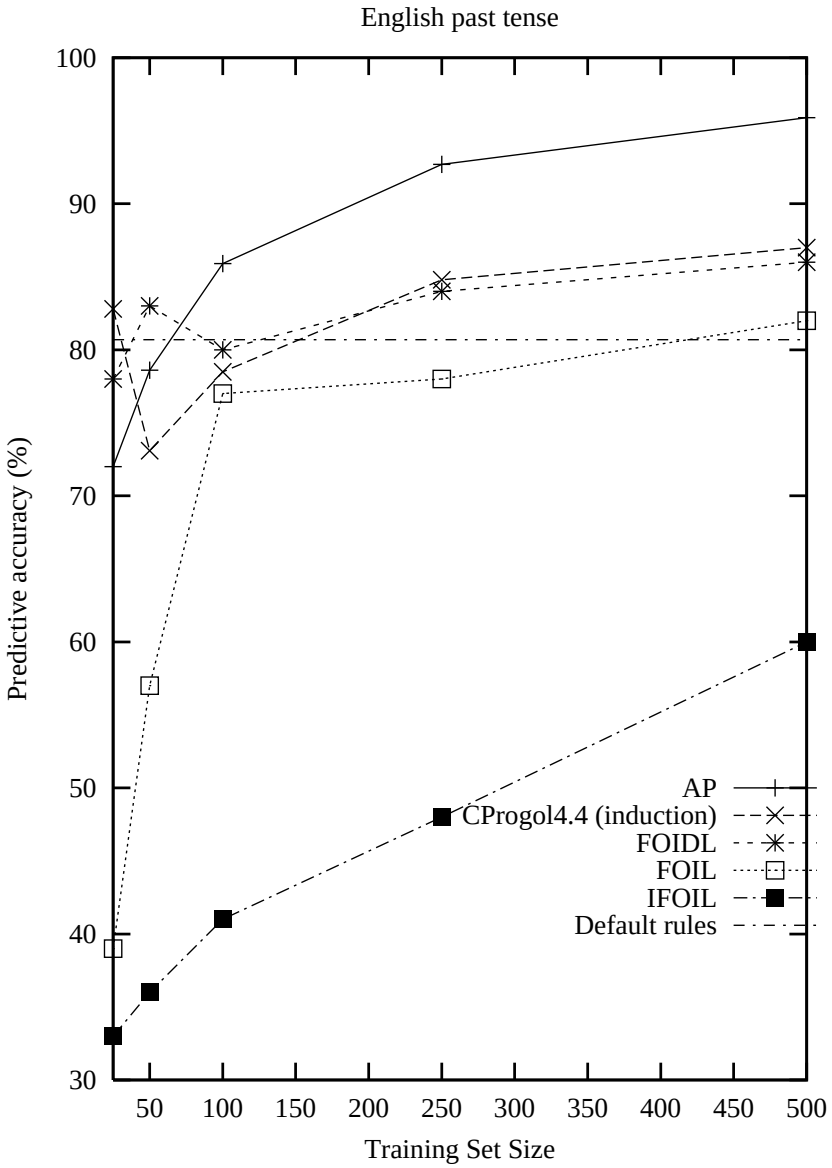

Fig. 5. Learning curves for alphabetic English past tense.

KRK illegality The two learning curves are shown in Figure 6. The horizontal line labelled "Majority class" shows the percentage of negative examples in the domain. Only the accuracy difference between induction and AP for 200 examples is significant at the 0.05 level, though taken together the differences are significant at the 0.0001 level. Thus null hypothesis 2 can be rejected.

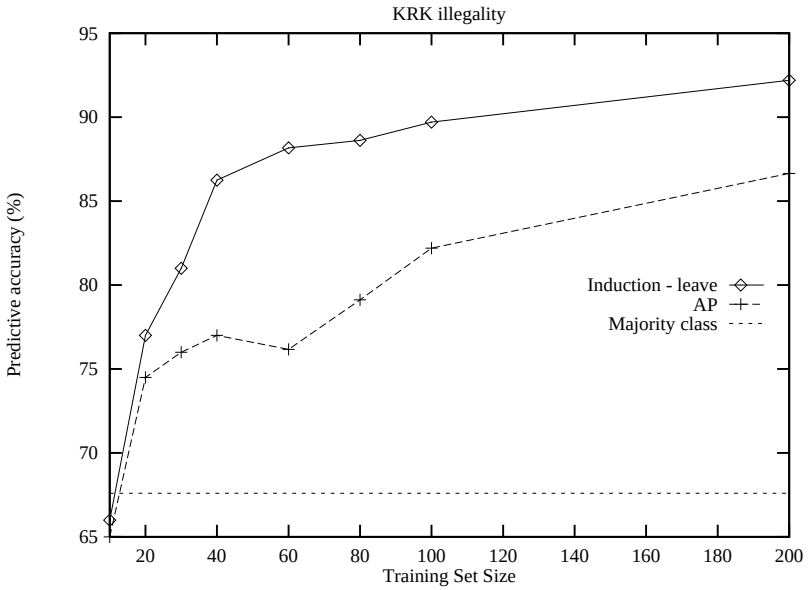

Fig. 6. Learning curves for KRK illegality.

5 Discussion and related work

The strong rejection of the two null hypotheses indicate that the advantages of AP relative to induction are domain dependent. The authors believe that AP has advantages for domains, like the English past tense, in which a large proportion of the examples must be treated as exceptions with respect to the hypothesis vocabulary. Note that KRK illegal contains exceptions, though they fall into a relatively small number of classes, and have relatively low frequency (a 2 clause approximation of KRK illegal has over 90% accuracy). By contrast, around 20% of the verbs in the past tense data are irregular.

It should be noted that our implementation of AP has a tendency to overgeneralise. This stems from the assymetry in constructing only clauses which make positive predictions in the *aleave* algorithm (Section 3). The tendency to overgeneralise decreases accuracy in the KRK illegal domain but increases accuracy in the past tense domain, due to the lack of negative examples. Even when

negative examples are added to the past tense training set, predictive accuracy is unaffected due to the output completeness constraint.

The AP accuracies on English past tense data shown in Figure 5 are the highest on this data set in the literature. However, it is interesting to note that CProgol's induction mode results are as good as FOIDL. This contradicts Mooney and Califf's claim that FOIDL's decision list representation gives FOIDL strong advantages in this domain.

5.1 Relationship of AP to analogy

Evans' [4] early studies of analogy concentrated on IQ tests of the form shown in Figure 7. Evans in was the first to implement a program for solving geometric

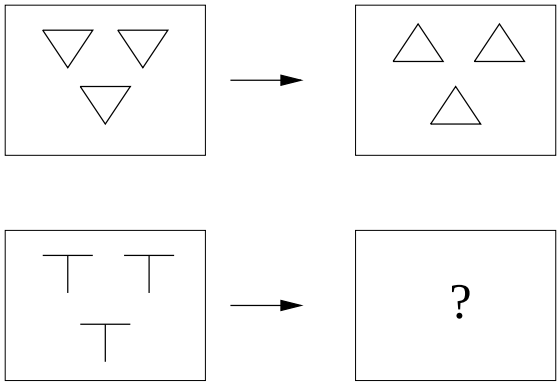

Fig. 7. IQ test problem of the type studied by Evans.

analogy problems. These are problems of the form "A is to B as C is to ?" where the answer is one of five possible solutions, i.e. a multiple-choice format. The problems solved by his program were taken from actual high-school level test papers. The program, called ANALOGY, comprised two parts. Part 1 is given two line drawings A and B as input and calculates a set of properties, relations and "similarities" such as rotations and reflections which take A into B and relate C to each of the five possible answers. Part 2 forms a set of theories or transformation rules taking A into B. It then attempts to generalize these theories to cover additional data (C and the answer figures). This results in a subset of the admissible theories, i.e. transformation rules which take A into B and C into exactly one answer figure. Finally, the program chooses the most specific theory from these admissible theories.

Evans notes that the program does very little search, which indicates that the hypothesis space of ANALOGY is highly constrained. The solution is chosen on the basis of a specificity bias. Although Part 1 of the program includes a large amount of domain-specific knowledge, Evans is careful to point out that Part 2 is a general purpose method for finding analogies of this form.

Analogical reasoning is often viewed as having a close relationship to induction. For instance, both Peirce and Polya suggested that analogical conclusions could be deduced via an inductive hypothesis [13, 14]. Similar views are expressed in the Artificial Intelligence literature [2]. For instance, Arima [1] formalises the problem of analogy as involving a comparison of a *base* B and *target* T. When B and T are found to share a similarity property S analogical reasoning predicts that they will also share a projected property P. This is formalised in the following analogical inference rule.

$$\frac{\begin{array}{l}P(B)\\S(T) \wedge S(B)\end{array}}{P(T)}$$

The rule above can be viewed as involving the construction of the following inductive hypothesis.

$$\forall x.S(x) \rightarrow P(x)$$

From this $P(T)$ can be inferred deductively. Note that S and P can obviously be extended to take more than one argument. For instance, given the Evans' type problem in Figure 7 we might formulate the following hypothesis as a Prolog clause.

```
is_to(X,Y) :- invertall(X,Y).
```

In this way we can view analogical reasoning as a special case of AP, in which the example set contains a single *base* example and the test instance relates to the *target*. According to Arima the following issues are seen as being central in the discussion of analogy.

1. What object (or case) should be selected as a base with respect to a target,
2. which property is significant in analogy among properties shared by two objects and
3. what property is to be projected w.r.t. a certain similarity.

These three issues are handled in the CProgol4.4 AP implementation as follows.

1. A set of base cases is used from the example set based on maximising compression over the hypothesis space,
2. relevant properties are found by constructing the bottom clause relative to the test instance and
3. the relevant projection properties are decided on the basis of modeh declarations given to CProgol.

6 Conclusions and further work

In this paper we have introduced the notion of AP as a half-way house between induction and instance-based learning. An implementation of AP has been incorporated into CProgol4.4

(ftp://ftp.cs.york.ac.uk/pub/ML_GROUP/progol4.4). In experiments AP produced the best predictive accuracy results to date on the English past tense data, outstripping FOIDL by around 10% after 500 examples. However, on KRK illegal AP performs consistently worse than CProgol4.4 in inductive mode. We believe that AP works best with domains in which a large proportion of the examples must be treated as exceptions with respect to the hypothesis vocabulary.

The present implementation of AP is limited in a number of ways. For instance, for any test instance x predictions must be binary, i.e. $y \in \{\text{True}, \text{False}\}$. Also, because no constructed hypotheses are ever stored, AP cannot deal with recursion. It is envisaged that a strategy which mixed both induction and AP might work better than either. Thus some, or all, of the AP hypotheses could be stored for later use. However, it is not yet clear to the authors which strategy would operate best.

Acknowledgements

The first author would like to thank Thirza and Clare for their good-natured support during the writing of this paper. This work was supported partly by the Esprit Long Term Research Action ILP II (project 20237), EPSRC grant GR/K57985 on Experiments with Distribution-based Machine Learning.

References

1. J. Arima. *Logical Foundations of Induction and Analogy.* PhD thesis, Kyoto University, 1998.
2. T. Davies and S.J. Russell. A logical approach to reasoning by analogy. In *IJCAI-87*, pages 264–270. Morgan Kaufmann, 1987.
3. W. Emde and D. Wettschereck. Relational Instance-Based Learning. In L. Saitta, editor, *Proceedings of the 13th International Machine Learning Conference*, pages 122–130, Los Altos, 1996. Morgan Kaufmann.
4. T.G. Evans. A program for the solution of a class of geoemtric analogy intellgence test questions. In M. Minsky, editor, *Semantic Information Processing.* MIT Press, Cambridge, MA, 1968.
5. A. Hutchinson. Metrics on terms and clauses. In M. Someren and G. Widmer, editors, *Proceedings of the Ninth European Conference on Machine Learning*, pages 138–145, Berlin, 1997. Springer.
6. C.X. Ling. Learning the past tense of english verbs: the symbolic pattern associators vs. connectionist models. *Journal of Artificial Intelligence Research*, 1:209–229, 1994.
7. R.J. Mooney and M.E. Califf. Induction of first-order decision lists: Results on learning the past tense of english verbs. *Journal of Artificial Intelligence Research*, 3:1–24, 1995.
8. S. Muggleton. Inverse entailment and Progol. *New Generation Computing*, 13:245–286, 1995.

9. S. Muggleton, M.E. Bain, J. Hayes-Michie, and D. Michie. An experimental comparison of human and machine learning formalisms. In *Proceedings of the Sixth International Workshop on Machine Learning*, Los Altos, CA, 1989. Kaufmann.
10. S. Muggleton and L. De Raedt. Inductive logic programming: Theory and methods. *Journal of Logic Programming*, 19,20:629–679, 1994.
11. S.H. Nienhuys-Cheng. Distance between Herbrand interpretations: a measure for approximations to a target concept. In N. Lavrač and S. Džeroski, editors, *Proceedings of the Seventh International Workshop on Inductive Logic Programming (ILP97)*, pages 321–226, Berlin, 1997. Springer-Verlag. LNAI 1297.
12. S.H. Nienhuys-Cheng. Distances and limits on Herbrand interpretations. In C.D. Page, editor, *Proceedings of the Eighth International Conference on Inductive Logic Programming (ILP98)*, pages 250–260, Berlin, 1998. Springer. LNAI 1446.
13. C.S. Peirce. Elements of logic. In C. Hartshorne and P. Weiss, editors, *Collected Papers of Charles Sanders Peirce*, volume 2. Harvard University Press, Cambridge, MA, 1932.
14. G. Polya. Induction and analogy in mathematics. In *Mathematics and Plausible Reasoning*, volume 1. Princeton University Press, Princeton, 1954.
15. D.E. Rumelhart and J.L. McClelland. On learning the past tense of english verbs. In *Explorations in the Micro-Structure of Cognition Vol. II*, pages 216–271. MIT Press, Cambridge, MA, 1986.