Avoiding noise fitting in a FOIL-like learning algorithm

Johannes Fürnkranz
juffi@ai.univie.ac.at
Austrian Research Institute for Artificial Intelligence
Schottengasse 3
A-1010 Vienna
Austria

Abstract

The research reported in this paper describes FOSSIL, an ILP system that uses a search heuristic based on statistical correlation. This algorithm implements a new method for learning useful concepts in the presence of noise. In contrast to FOIL's stopping criterion which allows theories to grow in complexity as the size of the training sets increase, we propose a new stopping criterion that is independent of the number of training examples. Instead, FOSSIL's stopping criterion depends on a search heuristic that estimates the utility of literals on a uniform scale.

1 Introduction

In this paper we introduce an Inductive Logic Programming algorithm closely related to FOIL [Quinlan, 1990]. FOSSIL uses a search heuristic based on statistical correlation. Advantages of this new heuristic are that there is no seperate calculation for negated literals and that the quality of literals is assessed on a uniform scale.

This paper is mainly concerned with the latter feature of this search heuristic. We show that it can advantageously be used to cut off all literals that have a heuristic value below a certain threshold. This eliminates the need for FOIL's encoding length stopping criterion. Experimental evidence supports our assumption that this method is successful in avoiding over-fitting the noise in the data and in learning useful concepts in the presence of noise. We show that FOSSIL converges towards a useful set of slightly over-general rules when increasing the size of the training set, while FOIL learns more and more complex concept descriptions that fit the noise in the training data.

Section 2 will give a short introduction into FOSSIL's search heuristic and section 3 will highlight the features of this heuristic we are mostly concerned with. Section 4 gives a short description of the setup used for the experiments reported in sections 5 and 6. In the last two sections we give a comparison to related work known from the ILP literature and we conclude.

2 FOSSIL's search heuristic

FOSSIL's evaluation function is based on the concept of statistical *correlation*. The *correlation coefficient* of two random variables X and Y is defined as

$$corr(X,Y) = \frac{E((X-\mu_X)(Y-\mu_Y))}{\sigma_X \times \sigma_Y} = \frac{E(X \times Y) - \mu_X \times \mu_Y}{\sigma_X \times \sigma_Y} \quad (1)$$

where μ and σ are *expected value* and *standard deviation*, respectively, of the random variables X and Y, and (see e.g. [Bosch, 1982]).

This *correlation coefficient* measures the degree of dependency of two series of points on a scale from -1 (*negative correlation*) to $+1$ (*positive correlation*). In the following description of its adaptation as a search heuristic for the Inductive Logic Programming algorithm FOIL, we will follow the notational conventions used in [Lavrač *et al.*, 1992].

Suppose FOSSIL has learned a partial clause c. Let the set of tuples T_c of size $n(c)$, containing $n^{\oplus}(c)$ positive and $n^{\ominus}(c)$ negative instances, be the current training set. We arbitrarily assign the numeric values $+1$ and -1 for the logical values *true* and *false*. The variable X in (1) now represents the multiset $V(c)$ of the signs (truth values) of the tuples in T_c. The variable Y denotes the multiset $V(L)$ of the truth values of a candidate literal L. A literal L is said to be *true*, whenever there exists a tuple in T_c that satisfies L; if L introduces new variables, they must have at least one instantiation that makes the literal true. Note that $V(c)$ and $V(L)$ naturally contain the same number of values.

The expected values in (1) will be estimated by the mean values of $V(c)$ and $V(L)$ respectively. Standard deviation will be approximated by the empirical variance. Thus we get

$$n = n(L) = n(c) = n^{\oplus}(c) + n^{\ominus}(c),$$

$$\mu_c = \frac{n^{\oplus}(c) - n^{\ominus}(c)}{n}, \mu_L = \frac{n^{\oplus}(L) - n^{\ominus}(L)}{n},$$

$$\sigma_c^2 = \frac{n^{\oplus}(c) \times (1-\mu_c)^2 + n^{\ominus}(c) \times (-1-\mu_c)^2}{n},$$

$$\sigma_L^2 = \frac{n^{\oplus}(L) \times (1-\mu_L)^2 + n^{\ominus}(L) \times (-1-\mu_L)^2}{n}$$

The last remaining term to be computed is $E(V(c) \times V(L))$. If both the truth values $v(c)$ and $v(L)$ of a tuple and the literal under scrutiny have the same sign, then $v(c) \times v(L) = 1$. Conversely, if one is positive and the other negative we have $v(c) \times v(L) = -1$. If we denote the number of positive tuples yielding a negative value for the literal L with $n^{\oplus}(c)^{\ominus}$ (and analogously define $n^{\oplus}(c)^{\oplus}$, $n^{\ominus}(c)^{\oplus}$ and $n^{\ominus}(c)^{\ominus}$), we get

$$E(V(c) \times V(L)) = \frac{n^{\oplus}(c)^{\oplus} + n^{\ominus}(c)^{\ominus} - n^{\ominus}(c)^{\oplus} - n^{\oplus}(c)^{\ominus}}{n}$$

The partial results of above now only need to be substituted into the formula for the correlation coefficient (1). As μ_c and σ_c only need to be evaluated once for each tuple set T_c, evaluation of this formula is not as complicated as it may seem at first sight. Also notice that with this approach no seperate calculation for negated literals has to be performed, as a high negative correlation indicates a high dependency on the negated literal.

The literal L_c with the highest absolute value of the correlation coefficient (or $\neg L_c$ if the sign of the coefficient is negative) is then chosen to extend c to form a new clause c'. This is based on the assumption that its high correlation with the current training set T_c indicates some form of causal relationship between the target concept and L_c. The set T_c is then extended to a new a set of tuples $T_{c'}$ (which in general will have a different size) and the process continues as described in [Quinlan, 1990].

3 Important features of the correlation coefficient heuristic

The information gain heuristic used in ID3 [Quinlan, 1983] and FOIL has been extensively compared to other search heuristics in decision tree generation [Mingers, 1989, Buntine and Niblett, 1992] and Inductive Logic Programming [Lavrač *et al.*, 1992]. The general consensus seems to be that it is hard to improve on this heuristic in terms of predictive accuracy in learning from noise-free data. While our results confirm this, we nevertheless claim that FOSSIL's evaluation function has some important features that distinguish it from the weighted information gain heuristic used in FOIL.

- In FOIL, the heuristic value of each literal and of its negation have to be calculated separately. FOSSIL does this in one calculation, as positive correlation indicates a causal relationship between the tuple set and the literal under scrutiny, while negative correlation indicates a causal relationship between the tuple set and the negation of the literal.

- The correlation between a tuple set and a determinate literal[1] is undefined, as σ_L will be 0 for determinate literals, because all tuples have at least one true variable assignment for this literal and thus $(1-\mu_L)$ and $n^{\ominus}(L)$ both will be 0. This allows the user to take care of the problem in a flexible way. The experiments reported in this paper ignored this problem by treating undefined cases as having correlation 0. Defining the heuristic value of determinate literals as 1 would put all determinate into the clause body. Irrelevant literals could be removed later in a post-processing phase [Quinlan, 1990]. Values between 0 and 1 result in the behavior proposed in [Quinlan, 1991]: Until a literal with a correlation above this pre-set value is found, determinate literals will be added to the clause body.

- The value of FOIL's evaluation function is dependent on the size of the tuple set. The same literal will have different information gain values in different example set sizes of the same concept, although its relative merit compared to its competitors will be about the same. FOSSIL's correlation coefficient on the other hand — after taking absolute values and choosing

[1] We say that a literal is *determinate* when it introduces a new variable that is always bound to exactly one value and thus yields no information gain (e.g. `plus( X, Y, Z)` with the new variable `Z`). This may cause problems, because the introduction of this new variable may be useful despite the fact that no information is gained.

the appropriate, positive or negative, literal — assigns a value on the uniform scale from 0 to 1. As the plausibility of a literal can now be judged on an absolute basis, the user can require the literals that are considered for clause construction to have a certain minimum correlation value. This can be used as a simple criterion for filtering out noise, as it can be expected that tuples originating from noise in the data will only have a small correlation with predicates in the background knowledge.

This paper reports experiments that confirm the last hypothesis.

4 Experimental setup

For the experiments in this paper we have used the domain of recognizing illegal chess positions in the KRK ending [Muggleton *et al.*, 1989], which has become a running example in ILP research. The goal is to learn the concept of an illegal white-to-move position with only white king, white rook and black king being on the board. The goal predicate is `illegal(A,B,C,D,E,F)` where the parameters correspond to the row and file coordinates of the pieces in the above order. Background knowledge consists of the predicates `X < Y`, `X = Y` and `adjacent(X,Y)`[2]. Typing constraints were used to speed up the search and recursion was not allowed for efficiency reasons.

Class noise in the training instances was generated according to the *Classification Noise Process* described in [Angluin and Laird, 1988]. In this model a noise level of η means that the sign of each example is reversed with a probability of η. Note that this differs from most of the results in the ILP literature, where a noise level of η means that, with a probability of η, the sign of each example is randomly chosen. Thus a noise level of η in our experiments is roughly equivalent to a noise level of 2η in the results reported in [Lavrač and Džeroski, 1992, Džeroski and Bratko, 1992b]. Noise was added incrementally, i.e. instances which had a reversed sign at a noise level η_1 also had a reversed sign at a noise level $\eta_2 > \eta_1$. Similarly, training sets with n examples were fully contained in training sets with $m > n$ examples.

In all experiments the induced rules were tested against sets of 5000 randomly chosen instances. It also proved useful to record the number of clauses in the induced concept and the average number of literals per clause to measure the complexity of the learned concept description.

5 The Cutoff

FOSSIL handles noise by simply not considering literals that have a correlation coefficient lower than a certain user-settable value — the *cutoff*. This results in a natural criterion for when to stop adding literals or clauses to the current concept definition. Whenever no literal has a correlation coefficient above the set threshold, the growing of the current clause stops and the examples that are

[2]Our definition of `adjacent` actually was `adjacent_or_equal`.

covered are removed from the tuple set.[3] If no literal above the cutoff can be found for starting a new clause, the current set of clauses is used as a concept definition. Note that it may happen that FOSSIL "refuses" to learn anything in cases where no predicate in the background knowledge has a significant correlation with the training data. This has actually happened several times, and is evident in the result with 20% Noise and and a Cutoff $C = 0.4$, where the average clause length is below 1 (see table 1).

We want to emphasize that this type of stopping criterion is not limited to FOSSIL's correlation coefficient heuristic, but may yield similar results with all search heuristics that assign values on a uniform scale, as e.g. the expected accuracy measure [Lavrač *et al.*, 1992].

The first series of experiments aimed at determining an appropriate value for this parameter for further experimentation. 10 training sets of 100 instances each were used at three different noise levels (5%, 10% and 20%). 6 different settings for the cutoff parameter C were used. The results averaged over the 5 runs are reported in table 1.

Noise		Cutoff					
		0.0	0.1	0.2	0.25	0.3	0.4
5%	Accuracy	93.05	93.05	93.32	93.58	95.57	93.86
	Clauses	6.3	6.3	6.2	5.8	4.2	2.7
	Lits/Clause	2.25	2.25	2.25	2.19	2.02	1.87
10%	Accuracy	87.77	87.77	90.0	93.44	93.52	83.18
	Clauses	8.2	8.2	6.3	4.5	3.8	1.8
	Lits/Clause	2.74	2.74	2.52	2.24	2.24	1.53
20%	Accuracy	80.21	80.21	85.21	86.87	87.00	72.48
	Clauses	11.4	11.4	6.0	4.1	3.2	0.7
	Lits/Clause	3.09	3.09	2.80	2.76	2.67	0.85

Table 1: Experiments with different settings for the *Cutoff*.

From these results the following observations can be made:

- A good setting for C in this domain seems to be somewhere around 0.3.
- There is a roughly linear transition from overfitting the noise to over-generalizing the rules. A low setting of C has a tendency to fit the noise, because most of the high correlation literals are above the threshold.[4] Conversely, a too optimistic setting of C results in over-generalization as too few literals have a correlation above the threshold.
- The complexity of the learned concepts ($\#Clauses \times \#Lits/Clause$) monotonically decreases with an increase of the cutoff parameter.
- The influence of a bad choice of the cutoff is more significant in data containing a larger amount of noise.

[3] Like FOIL, FOSSIL has a parameter that can enforce a given *minimum clause accuracy*, i.e. that a certain percentage of the examples covered by the clause must be positive.

[4] A setting of $C = 0$ results in learning a 100% correct rule for explaining the training set.

6 Comparison with FOIL

We performed two experiments to compare FOSSIL's performance to the performance of FOIL. In the first series we compared the behavior of the two systems with 10 training sets of 100 instances each at different noise levels, which has been the standard procedure for evaluating many ILP systems [Quinlan, 1990, Džeroski and Lavrač, 1991, Džeroski and Bratko, 1992b, Muggleton *et al.*, 1989]. In the second experiment we evaluated both programs at a constant noise level of 10%, but with an increasing number of training instances.

According to the results of the previous experiments we set $C = 0.3$ and never changed this setting.

6.1 Experiment 1

In this experiment we compared FOIL4 to FOSSIL at different noise levels. In order to have a fair comparison to FOSSIL where backtracking is not implemented, we used two versions of FOIL, regular FOIL4 and a new version, FOIL-NBT, where FOIL4's extensive mechanisms of backtracking and regrowing of clauses were not allowed. Surprisingly this version performed better than the original FOIL4 in noisy data as can be seen from the results of table 2.

Different Noise Levels		Noise							
		0%	5%	10%	15%	20%	25%	30%	50%
FOIL4	Accuracy	98.32	95.26	92.12	90.26	85.21	79.83	71.53	53.00
	Clauses	3.5	4.2	5.4	5.9	5.7	6.6	8.0	7.9
	Lits/Clause	1.64	1.98	2.41	2.47	2.66	2.98	3.03	3.45
FOIL-NBT	Accuracy	98.11	95.00	92.98	91.76	87.12	79.42	76.32	55.33
	Clauses	3.5	4.1	4.2	4.2	4.5	5.4	5.0	5.2
	Lits/Clause	1.64	1.98	2.34	2.48	2.67	2.80	2.79	3.08
FOSSIL (0.3)	Accuracy	98.54	95.57	93.52	92.83	87.00	81.63	70.59	(67.07)
	Clauses	3.7	4.3	3.8	4.2	3.2	2.7	0.7	0.0
	Lits/Clause	1.62	2.02	2.24	2.29	2.67	2.69	0.85	0.0

Table 2: A Comparison of FOIL and FOSSIL on different levels of noise.

An analysis of the result shows that FOSSIL performs best in most of the tests, but no significant difference between FOIL-NBT and FOSSIL can be found. A comparison of the average number of induced clauses and of the average literals per clause shows evidence that FOSSIL over-generalized at the high noise levels. A lower value of the cutoff parameter may result in better performance in the case of 30% noise, although it is unlikely that a useful theory would be learned. An interesting detail is that FOSSIL did not learn anything at a noise level of 50%, i.e. with totally random data. Thus the cutoff mechanism seems to be a primitive, but efficient means of distinguishing noise from useful information.

On the other hand, FOIL4 seems to perform worse than both, FOIL-NBT

and FOSSIL. The complexity of the concepts learned by FOIL4 increases with the amount of noise in the data, which is clear evidence for over-fitting noise in the data. Experiment 2 was designed to confirm this hypothesis.

6.2 Experiment 2

In this series of experiments we compared FOIL without backtracking to FOSSIL at different training set sizes, each having 10% noise. We decided to use FOIL-NBT instead of FOIL4, because it performed better at the previous series of tests. Besides, the version without backtracking naturally runs faster, which proved to be important. However, we have done a few sample runs with FOIL4 to confirm that its results would not be qualitatively different to those of FOIL-NBT.

Again, we used 10 different training sets and averaged the results. The outcomes of these experiments are summarized in table 3.

Different Training Set Sizes (10% Noise)		Training Set Size					
		100	250	500	750	1000	2000
FOIL-NBT	Accuracy	92.98	90.97	92.63	93.58	94.02	—
	Clauses	4.2	7.7	11.5	16.7	22.0	—
	Lits/Clause	2.34	3.31	3.61	3.89	4.15	—
FOSSIL (0.3)	Accuracy	93.52	92.68	92.79	96.33	98.05	98.41
	Clauses	3.8	3.7	3.1	3.0	3.0	3.0
	Lits/Clause	2.24	3.01	2.63	1.94	1.5	1.4

Table 3: A Comparison of FOIL and FOSSIL with different training set sizes

The most important finding is that FOIL clearly fits the noise, while FOSSIL avoids this and learns a slightly over-general, but much more useful theory instead. FOIL's fitting the noise has several disadvantages:

Accuracy: The more noisy examples there are in the training set, the more specialized are the various clauses in the concept description, which decreases the predictive ability of each clause. This results in an increasing difference between the over-all predictive accuracy of the rules learned by FOIL and FOSSIL.

Understandability: It is a widely acknowledged principle that the more complex a concept definition is, the less understandable it will be, in particular when both definitions describe the same data set. While the descriptions induced by FOIL for the large training sets were totally incomprehensible to the author, FOSSIL converged towards the simple, approximate theory of figure 1.[5] In fact in 8 of the 10 training sets with 2000 examples, precisely this theory was learned, while in the other two the literal `A \== C`

[5]This theory correctly classifies all but 4060 of the 262,144 possible domain examples (98.45%). 2940 positions (1.12%)with WK and WR on the same squares and 1120 positions (0.43%) where the WK is between WR and BK on the same row or file are erroneously classified [Fürnkranz, 1993]. (Remember that we have defined `adjacent` to mean `adjacent_or_equal`).

```
illegal(A,B,C,D,E,F) :- C = E.
illegal(A,B,C,D,E,F) :- D = F.
illegal(A,B,C,D,E,F) :- adjacent(A,E), adjacent(B,F).
```

Figure 1: An approximate theory that is 98.45% correct.

had been added to the first clause, which gives a 97.98% correct theory [Fürnkranz, 1993].

Efficiency: FOIL grows an increasing number of clauses with an increasing number of literals. Also, several of the literals chosen to fit the noise introduce new variables, which leads to an explosion of the size of the tuple set. In fact, the `C` implementation of FOIL could complete none of the ten experiments with 2000 training examples within 500 minutes of CPU time, while the PROLOG implementation of FOSSIL only needed about 15 minutes of CPU time for each of the training sets, running on the same machine.

What seems to be responsible for the drastic increase in the complexity of the learned clauses is that FOIL's stopping criterion [Quinlan, 1990] is dependent on the size of the training set. In the KRK domain it performs very well on sample sizes of 100 training examples. The more this number increases, the more bits are allowed for the theory to explain the data. However, more examples do not necessarily originate from a more complex theory. In fact, FOIL very often chooses the same literals as FOSSIL for the first clauses of its concept definition, but then continues to add literals and clauses, where FOSSIL stops.

FOSSIL uses a statistical stopping criterion based on the assumption that each literal in an explanation must have a significant correlation with the set of training examples. Statistical measures usually improve with the size of the training sets and so does the quality of the rules induced by FOSSIL. While both FOIL and FOSSIL successively improve their predictive accuracy with increasing training set sizes, only FOSSIL converges towards a useful theory.

7 Related Work

A comparison of the above findings to the relevant results reported for mFOIL [Džeroski and Bratko, 1992b] and LINUS [Lavrač and Džeroski, 1992] would be interesting, but the results cited for the performance of FOIL differ in all these papers. Considering the different noise model we are using, our results for FOIL are significantly better than in both other papers. The reason for this might be our different definition of the `adjacent`-relation. However, none of the above papers reports the complexity of the learned clauses, and none of them performs experiments with increasing training set sizes. The architecture of FOSSIL resembles that of mFOIL, but while mFOIL uses a parameter to adjust the degree to which the examples can be trusted, FOSSIL's cut-off parameter can be used to trade off over-generality and over-fitting. In

addition, mFOIL and FOIL both have to do additional calculations to determine when to stop learning — mFOIL computes a statistical significance test [Džeroski and Bratko, 1992a], while FOIL uses a heuristic based on the compression of the theory [Quinlan, 1990]. FOSSIL's simple cutoff method reduces the amount of additional computation to a mere comparison. We currently work on repeating the test series described above on mFOIL and on a version of FOIL that uses a similar cutoff stopping criterion.

[Srinivasan *et al.*, 1992] report a series of similar experiments using CW-GOLEM. Here the results are directly comparable to our findings, as the same noise model has been used in both experiments. In the presence of 10% noise CW-GOLEM also converges towards the approximate theory of figure 1, and it seems to converge faster towards this theory than FOSSIL. However, CW-GOLEM is first generating a highly specialized theory, from which the most compressive set of clauses is selected in a post-processing phase, while FOSSIL bases its pruning decisions solely on heuristic estimates without any look-ahead.

8 Conclusion

In this research we have shown that FOIL's stopping criterion is not appropriate for learning in noisy domains, as it allows for fitting the noise in the examples. To circumvent this problem we have proposed a different search heuristic based on statistical correlation which yields values on a uniform scale. This allows to assess the heuristic value of a certain literal on an absolute basis, and thus to drop literals with values below a certain threshold. This method of handling noise proved especially useful with increasing training set sizes.

We believe that search heuristics based on quality estimates on a uniform scale can be used advantageously in several other ways, e.g. for a simple best-first or beam search in the space of literals, or for an incremental version of the algorithm. This remains to be explored in the near future. We also want to perform experiments in different domains to see whether the cutoff parameter, whose setting has so far been experimentally determined, has some domain independence. There is some evidence that the optimal value of the cutoff parameter changes with the amount of noise in the theory, so a dynamical change or automatic assessment of the parameter during learning seems to be another promising field for further research.

Acknowledgements

This research is sponsored by the Austrian *Fonds zur Förderung der Wissenschaftlichen Forschung (FWF)* under grant number P8756-TEC. Financial support for the Austrian Research Institute for Artificial Intelligence is provided by the Austrian Federal Ministry of Science and Research. I would like to thank J. R. Quinlan and R. M. Cameron-Jones for making FOIL4 `ftp`-able and Gerhard Widmer for providing a PROLOG implementation of the FOIL algorithm to start with and for encouraging this research.

References

[Angluin and Laird, 1988] D. Angluin and P. Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1988.

[Bosch, 1982] Karl Bosch. *Elementare Einfűhrung in die angewandte Statistik*. Friedr. Vieweg & Sohn, Braunschweig/Műnchen, 2nd edition, 1982.

[Buntine and Niblett, 1992] Wray Buntine and Tim Niblett. A further comparison of splitting rules for decision-tree induction. *Machine Learning*, 8:75–85, 1992.

[Džeroski and Bratko, 1992a] Sašo Džeroski and Ivan Bratko. Handling noise in Inductive Logic Programming. In *Proceedings of the International Workshop on Inductive Logic Programming*, Tokyo, Japan, 1992.

[Džeroski and Bratko, 1992b] Sašo Džeroski and Ivan Bratko. Using the m-estimate in Inductive Logic Programming. In *Logical Approaches to Machine Learning, Workshop Notes of the 10th European Conference on AI*, Vienna, Austria, 1992.

[Džeroski and Lavrač, 1991] Sašo Džeroski and Nada Lavrač. Learning relations from noisy examples: An empirical comparison of LINUS and FOIL. In *Proceedings of the 8th International Workshop on Machine Learning*, pages 399–402, Evanston, Illinois, 1991.

[Fürnkranz, 1993] Johannes Fürnkranz. A numerical analysis of the KRK domain. Working Note, 1993. Available upon request.

[Lavrač and Džeroski, 1992] Nada Lavrač and Sašo Džeroski. Inductive learning of relations from noisy examples. In Stephen Muggleton, editor, *Inductive Logic Programming*, pages 495–516. Academic Press Ltd., London, 1992.

[Lavrač *et al.*, 1992] Nada Lavrač, Bojan Cestnik, and Sašo Džeroski. Search heuristics in empirical inductive logic programming. In *Logical Approaches to Machine Learning, Workshop Notes of the 10th European Conference on AI*, Vienna, Austria, 1992.

[Mingers, 1989] John Mingers. An empirical comparison of selection measures for decision-tree induction. *Machine Learning*, 3:319–342, 1989.

[Muggleton *et al.*, 1989] Stephen Muggleton, Michael Bain, Jean Hayes-Michie, and Donald Michie. An experimental comparison of human and machine learning formalisms. In *Proceedings of the 6th International Workshop on Machine Learning*, pages 113–118, 1989.

[Quinlan, 1983] J. Ross Quinlan. Learning efficient classification procedures and their application to chess end games. In Ryszard S. Michalski, Jaime G. Carbonell, and Tom M. Mitchell, editors, *Machine Learning. An Artificial Intelligence Approach*, pages 463–482. Tioga Publishing Co., 1983.

[Quinlan, 1990] John Ross Quinlan. Learning logical definitions from relations. *Machine Learning*, 5:239–266, 1990.

[Quinlan, 1991] John Ross Quinlan. Determinate literals in inductive logic programming. In *Proceedings of the 8th International Workshop on Machine Learning*, pages 442–446, 1991.

[Srinivasan *et al.*, 1992] A. Srinivasan, S. H. Muggleton, and M. E. Bain. Distinguishing noise from exceptions in non-monotonic learning. In *Proceedings of the International Workshop on Inductive Logic Programming*, Tokyo, Japan, 1992.