A Comparison of ILP and Propositional Systems on Propositional Traffic Data

Sam Roberts
OUCL,
Wolfson Building,
Parks Road,
Oxford OX1 3QD
Samuel.Roberts@comlab.ox.ac.uk

Nico Jacobs, Wim Van Laer
Katholieke Universiteit Leuven,
Department of Computer Science,
Celestijnenlaan 200 A,
B-3001 Heverlee,
Belgium
Nico.Jacobs@cs.kuleuven.ac.be
Wim.VanLaer@cs.kuleuven.ac.be

Stephen Muggleton
Department of Computer Science,
University of York,
York YO1 5DD
stephen@cs.york.ac.uk

Jeremy Broughton
Transport Research Laboratory,
Old Wokingham Road,
Crowthorne,
Berks RG45 6AU
JeremyB@E.TRL.CO.UK

March 16, 1998

Abstract

This paper presents an experimental comparison of two Inductive Logic Programming algorithms, Progol and Tilde , with C4.5, a propositional learning algorithm, on a propositional dataset of road traffic ac-

cidents. Methods are described for handling the skewed distribution of positive and negative examples in this dataset, and the relative cost of errors of commission and omission in this domain. Rules learnt by the algorithms are evaluated by these methods, and the conclusion is drawn that ILP algorithms can perform competitively on such a propositional domain.

1 Introduction

ILP (Inductive Logic Programming) has been defined [10] as the intersection of Machine Learning and Logic Programming, and as such it represents an advance over propositional learning systems. ILP systems are able to handle complex relational data which are not representable propositionally, such as chemical structures (e.g. see [16]). However, there are many domains of interest which are representable propositionally, and ILP systems must also be able to handle these domains as effectively as a propositional learner if they are to become as widely used.

This paper presents an experimental comparison of two Inductive Logic Programming systems, PROGOL and TILDE , with C4.5, a standard propositional algorithm, on just such a domain, that of traffic accidents. The paper grew from previous work [17] and from applications of various systems to this data at an ILP Transport workshop held at the University of York on December 3rd, 1997.

2 Systems

2.1 PROGOL

PROGOL is a state of the art ILP system described in [11]. The user specifies a restricted language of first order expressions to be used as the hypothesis space H. Restrictions are stated using "mode declarations", which specify which predicates are to be used in the heads and bodies of learnt rules, and the types and formats of their arguments.

PROGOL uses a sequential covering algorithm to learn a set of rules from H which cover the examples. For each positive example e^+ not yet covered, PROGOL constructs the most specific hypothesis $\perp$ from H such that $B \wedge \perp \vdash e^+$, where B is background knowledge supplied. PROGOL then performs a general to specific search of the hypothesis space bounded by $\perp$. During this search, PROGOL looks for the hypothesis having minimum description length.

PROGOL has been used successfully in many domains, including finite element mesh design [8], mutagenesis [16], natural language [6] and road traffic accidents [17]. It is available by anonymous ftp from ftp.cs.york.ac.uk in the directory pub/ML_GROUP/progol4.4 and includes a manual written by the author.

2.2 TILDE

A main feature of TILDE [2, 3] is the representation of examples, which correspond to a small relational database (or Prolog knowledge base). In other words, an example consists of multiple relations and each example can have multiple tuples for these relations. This setting is known in the literature as *learning from interpretations* and was first introduced in [15]. This representation is a natural upgrade of the attribute-value representation where each example consists of a single tuple in a relational database. *Learning from interpretations* contrasts with the classical inductive logic programming setting *learning from entailment* which is employed by systems such as PROGOL and FOIL . Details about the relation between these different settings can be found in [14].

The setting *learning from interpretations* allows us to upgrade a propositional learner towards a relational learner. TILDE is a first-order upgrade of an existing attribute-value learning system, Quinlan's popular predictive C4.5 algorithm for decision tree induction [13]. Other examples of first-order upgrades are ICL , which upgrades the predictive production rule approach as incorporated in e.g. CN2 [5, 4] and AQ [9], and WARMR [7], which extends APRIORI [1] to mine association rules in multiple relations.

TILDE learns a theory which discriminates as well as possible between (training) examples of different classes. The learned theory can be used to classify new, unseen examples into one of the available classes.

TILDE (Top down Induction of Logical DEcision trees) builds a decision tree similar to C4.5, but with first order logic conjunctions of literals as tests in the nodes. Heuristics are used to choose the best test at each node. The resulting classification tree can also be outputted as a Prolog program or a logical program. TILDE can discretize numerical values and has look ahead ability (see [2]). TILDE is an efficient ILP-system that can handle large datasets (in experiments datasets up to 100 Mb were used).

More information on TILDE and ICL can be found at the following URL:

`http://www.cs.kuleuven.ac.be/~ml/MLRG-E.shtml`

2.3 C4.5

C4.5 [13] is an extension of Quinlan's ID3 system [12]. Decision trees are induced by growing them from the root downward, using a greedy algorithm to select the next best attribute for each new decision branch added to the tree. The complete hypothesis space is searched, with an inductive bias to prefer smaller trees. Extensions to ID3 include methods for avoiding overfitting data and for handling training examples with missing attribute values.

3 Data and Datasets

3.1 The Road Accident Reporting System in Great Britain

In Great Britain, the national database of police accident reports contains details of all road accidents involving personal injury of which the police become aware within 30 days. Each report contains:

- 1 Attendant Circumstances Record
- 1 Casualty Record for each injured person
- 1 Vehicle Record for each vehicle involved

The data are largely objective, such as road number or age of driver, and do not include details of why the accidents occurred. The potential contribution of such information to attempts to improve road safety has been recognised for several years, and recent research at the Transport Research Laboratory has developed a new system for recording these 'Contributory Factors'. It is hoped that this system will be adopted as a regular part of the national reporting system.

The new system was trialed with 8 police forces for 3 months in 1996, and the data analysed in this paper is the result of matching about half of the Contributory Factor reports with the accident records routinely collected by the police.

The new system is unique in dividing the factors into two groups: 'What went wrong' (known as Precipitating Factors) and 'Why?' (known as Contributory Factors). This approach leads reporting officers to structure their investigations: they work back from the actual accident to identify the principal failure or manoeuvre which led directly to the accident, select the appropriate Precipitating Factor from the list, then try to establish the reasons for this failure or manoeuvre. Up to four Contributory Factors can be entered, in order of diminishing importance, and each is marked as Definite, Probable or Possible. The trial showed that the police were able to operate the system with little training, and the data collected were of good quality.

3.2 Dataset

The dataset used for this study consists of 1413 accident reports, during the period from April until August 1996. It was studied previously in [17] to find rules for predicting accidents caused by young male drivers (a class of accidents of interest to traffic experts).

Accidents are given preclassified as positive or negative. Positive examples are those accidents which were caused by a young male driver.

Each of the above types of record (attendant circumstances, casualties, vehicles, and contributory factors) is represented by the following four predicates.

- `acc_record/25`

- `cas_record/24`
- `veh_record/16`
- `caus_record/19`

Seventeen fields were selected from these records for relevance to classifying accidents as caused by young male drivers, by experience from the previous study. The values of these fields for each accident form the background knowledge concerning the accidents.

4 Experimental Description

4.1 Learning task

The dataset of 1413 examples was split 10 times randomly and independently into training and test sets. The training sets contained 70% of the examples and the corresponding test set contained the remaining 30%. A second set of training sets was also constructed from the first by rebalancing (see section 4.2).

Using each algorithm in turn, and supplying identical background knowledge in each case, rules for predicting whether an accident was caused by a young male driver were learnt from each training set, and from each rebalanced training set. The time taken to learn was recorded, and the accuracy of the learnt rules was assessed on the corresponding test set.

4.2 Rebalancing

The data under discussion in this paper have two unusual properties. The first is that they are skewed towards negative examples; out of 1413 examples, only 255 are positive and 1158 negative. The second is that traffic experts are less concerned about errors of commission than about errors of omission; it is more important to be able to predict when an accident *is* caused by a young male driver than when it is not. These properties, when combined, reduce the value of standard testing methods; accuracies and χ^2 figures obtained from contingency tables give too much weight to negative predictions. What is needed is a method of reweighting the influence of the positive examples on these results.

This effect can be reduced by rebalancing the training and test data. A training set can be rebalanced by the following method. All positive examples in the set are kept, along with an equal number of the negative examples in the set, randomly selected. Thus a rebalanced training set consists of 50% positive and 50% negative examples.

A test set can be rebalanced in a similar way, or the effect can be simulated easily by the following method. Say a set of rules has been tested on an unrebalanced test set T, giving the contingency table on the left of table 1. We can construct a new contingency table from this by multiplying the left hand

	Actual	
Predicted	Positive	Negative
Positive	A	B
Negative	C	D

	Actual	
Predicted	Positive	Negative
Positive	$\frac{A}{2P}$	$\frac{B}{2(1-P)}$
Negative	$\frac{C}{2P}$	$\frac{D}{2(1-P)}$

Table 1: Reweighting a contingency table to simulate a rebalanced test set. Here $P = \frac{A+C}{A+B+C+D}$.

Algorithm	Original	Rebalanced
PROGOL	80.78% ± 0.61%	55.25% ± 0.76%
TILDE	80.93% ± 0.60%	55.25% ± 0.76%
C4.5	81.64% ± 0.60%	54.49% ± 0.77%
Majority Class	81.95%	50%

Table 2: Evaluation of rules learnt from original training sets on original and rebalanced test sets.

column by $\frac{1}{2P}$ and the right hand column by $\frac{1}{2(1-P)}$, where

$$P = \frac{A+C}{A+B+C+D}.$$

This ratio ensures that the totals for both contingency tables remain equal, and the new table gives the values which we would have obtained had we rebalanced T. Note that "accuracy" values obtained from the resulting table cannot correctly be regarded as such; rather they are an estimate of how valuable the predictions will be to an expert who places a higher cost on errors of omission than commission.

5 Results

The results shown in tables 2 and 3 show evaluations of rules learnt from original and rebalanced training sets. Note again that only figures for original training sets on original test sets can be regarded as accuracy values. The majority class (empty algorithm) is also shown.

Table 4 shows the mean times taken to learn from original and rebalanced training sets. Standard deviations are given in brackets where available. All times were measured on an i586 running Linux2.0.30.

Algorithm	Original	Rebalanced
PROGOL	66.79% ± 0.72%	58.42% ± 0.76%
TILDE	66.08% ± 0.73%	61.72% ± 0.75%
C4.5	68.47% ± 0.71%	62.57% ± 0.74%
Majority Class	50%	50%

Table 3: Evaluation of rules learnt from rebalanced training sets on original and rebalanced test sets.

	Learning time (secs)	
Algorithm	Original	Rebalanced
PROGOL	2249.7 (93.39)	579.4 (42.32)
TILDE	585.0 (98.85)	95.0 (13.54)
C4.5	3.09	0.8

Table 4: Mean times taken for learning from each training set. Figures in brackets are standard deviations.

6 Discussion

The clearest implication of the results is the difference in times taken for learning between the systems. TILDE is significantly faster than PROGOL , and C4.5 is significantly faster than either of the two ILP algorithms.

What is more interesting is the information in tables 2 and 3. Let us first consider rules learnt from original training sets (table 2). When evaluated normally, C4.5 gives a greater accuracy than PROGOL and TILDE , though since the error bars overlap, this is not significant. However, when evaluated on a rebalanced test set to give a more realistic appraisal of rules from the point of view of a traffic expert, the reverse is the case. The explanation for this is that the proportion of C4.5's correct predictions which were negative was much higher than that of the ILP systems. Note also that when avaluated normally, all algorithms perform worse than the majority class algorithm. When evaluated on rebalanced data, the majority class algorithm performs significantly more poorly.

The situation with rules learnt from rebalanced training sets is not so clear cut, but nevertheless points in a similar direction. When evaluated on the original test sets, C4.5 performs better than the ILP systems. PROGOL does drop a little, but C4.5's lead over TILDE is much reduced.

These results suggest that ILP systems, as well as being able to cope with a wider range of more structurally complex domains, can also compete on an even footing with propositional learning algorithms on their home ground.

7 Conclusions

This paper has presented an experimental comparison of three learning algorithms, two ILP and one propositional, on a propositional dataset. It has shown that, whilst being significantly slower, ILP systems can indeed perform as well as propositional learning systems in such a domain. This is especially true when methods of evaluation are used which reflect the understanding and values of domain experts more appropriately than simple contingency table analysis.

8 Acknowledgements

Sam Roberts would like to thank Smith System Engineering for their support during this work, Stefan Wrobel for his work on MIDOS which unfortunately could not be included, and Stephen Muggleton for his help and encouragement. This work was supported by the Esprit Long Term Research Action ILP II (project 20237), EPSRC grant GR/K57985 on Experiments with Distribution-based Machine Learning, an EPSRC Case award held by Sam Roberts and an EPSRC Advanced Research Fellowship held by Stephen Muggleton. Nico Jacobs is supported by the Flemish Institute for the Promotion of Scientific and Technological Research in the Industry (IWT). Wim Van Laer is supported by the Fund for Scientific Research, Flanders. Nico Jacobs and Wim Van Laer would like to thank Hendrik Blockeel, Luc Dehaspe and Luc De Raedt for their valuable comments and feedback.

References

[1] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A.I. Verkamo. Fast discovery of association rules. In *Advances in Knowledge Discovery and Data Mining*, pages 307–328. MIT Press, 1996.

[2] H. Blockeel and L. De Raedt. Lookahead and discretization in ILP. In *Proceedings of the 7th International Workshop on Inductive Logic Programming*, volume 1297 of *Lecture Notes in Artificial Intelligence*, pages 77–85. Springer-Verlag, 1997.

[3] H. Blockeel and L. De Raedt. Top-down induction of first order logical decision trees. Submitted, 1998.

[4] P. Clark and R. Boswell. Rule induction with CN2: Some recent improvements. In Yves Kodratoff, editor, *Proceedings of the 5th European Working Session on Learning*, volume 482 of *Lecture Notes in Artificial Intelligence*, pages 151–163. Springer-Verlag, 1991.

[5] P. Clark and T. Niblett. The CN2 algorithm. *Machine Learning*, 3(4):261–284, 1989.

[6] J. Cussens, D. Page, S. Muggleton, and A. Srinivasan. Using Inductive Logic Programming for Natural Logic Processing. In W. Daelemans, T. Weijters, and A. van der Bosch, editors, *ECML'97 – Workshop Notes on Empirical Learning of Natural Language Tasks*, pages 25–34, Prague, 1997. University of Economics. Invited keynote paper.

[7] L. Dehaspe and L. De Raedt. Mining association rules in multiple relations. In *Proceedings of the 7th International Workshop on Inductive Logic Programming*, volume 1297 of *Lecture Notes in Artificial Intelligence*, pages 125–132. Springer-Verlag, 1997.

[8] B. Dolsak and S. Muggleton. The application of Inductive Logic Programming to finite element mesh design. In S. Muggleton, editor, *Inductive Logic Programming*, pages 453–472. Academic Press, London, 1992.

[9] R.S. Michalski. A theory and methodology of inductive learning. In *Machine Learning: an artificial intelligence approach*, volume 1. Morgan Kaufmann, 1983.

[10] S. Muggleton. Inductive logic programming. *New Generation Computing*, 8(4):295–318, 1991.

[11] S. Muggleton. Inverse entailment and Progol. *New Generation Computing*, 13:245–286, 1995.

[12] J.R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[13] J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993. Morgan Kaufmann series in machine learning.

[14] L. De Raedt. Induction in logic. In R.S. Michalski and J. Wnek, editors, *Proceedings of the 3rd International Workshop on Multistrategy Learning*, pages 29–38, 1996.

[15] L. De Raedt and S. Džeroski. First order *jk*-clausal theories are PAC-learnable. *Artificial Intelligence*, 70:375–392, 1994.

[16] A. Srinivasan, S. Muggleton, R. King, and M. Sternberg. Theories for mutagenicity: a study of first-order and feature based induction. *Artificial Intelligence*, 85(1,2):277–299, 1996.

[17] B. Williams, S. Roberts, and S. Muggleton. Use of ILP to investigate accident data: final report. Technical Report JA022D010-0.1, Smith System Engineering, Surrey Research Park, Guildford, Surrey GU2 5YP, September 1997.