Invented Predicates to Reduce Knowledge Acquisition Effort

Hendra Suryanto and Paul Compton

Artificial Intelligence Department
School of Computer Science and Engineering,
The University of New South Wales Sydney 2052, Australia

email: {hendras, compton}@cse.unsw.edu.au

UNSW-CSE-TR-0302
March 2003

Abstract

The aim of this study was to develop machine learning techniques that would speed up knowledge acquisition from an expert. As the expert provided knowledge the system would generalize from this knowledge in order to reduce the need for later knowledge acquisition. This generalization should be completely hidden from the expert. We have developed such a learning technique based on Duce's intra construction operator and absorption operators (Muggleton, 1990) and applied to Ripple Down Rules (RDR) incremental knowledge acquisition (Compton & Jansen, 1990). Preliminary evaluation shows that knowledge acquisition can be reduced by up to 50%.

1. Introduction

RDR was developed to deal with the problem that experts never give a comprehensive explanation for their decision making. Rather they justify that the conclusion is correct and the justification is created for and shaped by the context in which it is given (Compton & Jansen 1990)

The critical features of RDR are that:

- Knowledge is added to the knowledge base (KB) to deal with specific cases. These are cases for which the system has made an error.
- The KB gradually evolves over time, whilst in use dealing with real cases.
- The system rather than the knowledge engineer or expert organizes the structure of the KB.
- Any knowledge acquisition is validated so that the knowledge added provides an incremental addition to the system's knowledge and does not degrade previous knowledge.
- To add new knowledge, the expert only has to identify features in a case that distinguish it from other related cases retrieved by the system.
-

RDR systems have been implemented for a range of application areas and tasks. The first industrial demonstration of this approach was the PEIRS system, which provided clinical interpretations for reports of pathology testing (Edwards, et al., 1993). The approach has also been adapted to a of tasks: multiple classification (Kang, Compton & Preston, 1995), control (Shiraz & Sammut 1997) heuristic search (Beydoun & Hoffman, 1997; Beydoun & Hoffman, 1998), document management using multiple classification (Kang, et al. 1997), configuration (Compton, et al., 1998) and resource allocation (Richards & Compton, 1999). Generalized first-order RDR have also been proposed (Drake & Beydoun 2000). The level of evaluation in these studies varies, but overall they demonstrate very simple and highly efficient knowledge acquisition.

RDR tools to build KBs which provide interpretative comments for medical Chemical Pathology reports are now available commercially and at least 50 Kbs have been developed by various laboratories and are in routine clinical use. Results from this experience have not yet been published, but confirm that very large knowledge bases (>7000 rules) can be built and maintained by pathologists with little computing experience or knowledge. On average it takes about one minute to add a rule. A complete training course before pathologists start building and maintaining such systems takes less than one day (Pacific Knowledge Systems, personal communication). RDR systems have also been shown to converge and end up similar sized KBs to those developed by machine learning techniques (Compton, et al., 1994; Kang, Compton & Preston, 1998) and cannot be compressed much by simple reorganization (Suryanto, Richards & Compton, 1999)

The exception structure of RDR has also been found to be a useful representation for machine learning where it tends to produce more compact KB than other representations (Catlett, 1992). Various machine learning RDR systems have been developed (Gaines & Compton, 1992; Kivinen, Manila & Ukkonen, 1993; Scheffer, 1996; Siromoney & Siromoney, 1993; Wada, Motoda & Washio, 2000). However, the machine learning systems related to RDR do not replace RDR intended for use by an expert. As always machine learning systems depend on well classified examples in sufficient numbers. An expert building an RDR KB can provide a rule, and a working system will start to evolve, when classifying a case. Experts can deal successfully with single rare cases and the commercial RDR pathology knowledge bases appear to have a wide range of subtle comments.

2. Aim

RDR reduce the need for knowledge engineering, but they do not remove it completely. In most domains the expert will prefer to construct rules about higher level features rather than the actual raw data. Some initial knowledge engineering is required to set up the appropriate feature extraction for the expert to use. The expert then builds rules that go straight from initial features to final conclusion; there is no intermediate structure.

This lack of internal structure has been a criticism of RDR in the knowledge acquisition community. One of the landmark papers in this community is Clancey's paper on heuristic classification (Clancey, 1985), which observed that the majority of classification systems could be characterized as having rules that produced coarse intermediate conclusions and that these were then refined into detailed final conclusions.

The partial response to these criticisms is that despite having a flat structure it is quicker and easier to build and maintain an RDR system. RDR were first developed to deal with the difficulty of adding rules to systems to a heuristic classification structure (Compton et al., 1989).

The ideal solution would be a system with intermediate conclusions but the same ease of knowledge acquisition. Intermediate conclusions that the expert can define were introduced into RDR (Beydoun & Hoffmann 2000), (Compton & Richards 2000) and are also used in the commercial RDR systems. However, this is unattractive as it shifts the expert task from simply identifying reasons for a conclusion, to a knowledge engineering task of structuring knowledge.

The aim of this work then is to develop a method of predicate invention that can be used in conjunction with RDR to automatically discover intermediate conclusions. Such a method not only needs to discover possible intermediate conclusions, but assess their usefulness and remove those that are problematic. This should all be hidden from the expert, so that as far as they are concerned, they simply identify features in a case as to why it should have a particular conclusion.

3. Method

3.1 Predicate invention

Matheus and Rendell (1989) defined feature construction as the application of set of constructive operators {o1, o2, ...on} to a set of existing features {f1, f2, ...fm}, resulting in the construction of one or more new features {f1', f2', ...fN'} intending for use in describing the target concept. Bloedorn and Michalski (1991) have investigated feature construction where operators are algebraic such as equal, addition, substraction, multiplication etc. which are applied to pair of existing features. Zupan et al. (1997) proposed a suboptimal heuristic algorithm to decompose existing functions (in the RDR context such functions are rules). This function decomposition is data-driven and uses compactness and accuracy of the resulting rule set as a success measure. Sutton and Matheus (1991) have investigated feature constructions for learning higher polynomial functions from examples. ILP researchers take a further step in constructing first order predicates (Srinivasan & King, 1996) and (Flach & Lavrac, 2000).

In our present study we have to limited the constructive operators to the Boolean operators AND, OR and NOT, dealing with the predefined features available in the KBS. This applies to RDR, but also to rule-based systems in general. We restricted the operators because our main concern was integration with knowledge acquisition to produce a heuristic classification-like system.

The closest to our goals is the Duce system (Muggleton, 1990) which suggests possible higher level domain features to the expert. In our application the higher level features are hidden from the expert and our goal is to reduce the number of interactions with the expert. Duce employs six operators: inter-construction, intra-construction, absorption, identification, dichotomisation and truncation. We use only the intra-construction and absorption operators.

The application of the intra-construction operator is as follows:

$X \leftarrow B, C, D, E$ (1)
$X \leftarrow A, B, D, F$ (2)
are replaced by:
$X \leftarrow B, D, Z$ (3)
$Z \leftarrow C, E$ (4)
$Z \leftarrow A, F$ (5)
where Z is an invented predicate or intermediate conclusion. The absorption operator, first proposed Sammut and Banerji (1986) is used as follows:
$X \leftarrow A, B, C, D, E,$ (6)
$Y \leftarrow A, B, F$ (7)
are replaced by:
$X \leftarrow A, B, D, Z,$ (8)
$Y \leftarrow B, Z$ (9)
using the new predicate Z from (4) and (5).

3.2 Algorithm

The following algorithm describes how the Duce operators are integrated into RDR knowledge acquisition. However, one cannot simply use the Duce operators alone as there has to be some assessment of the quality of any invented predicate. Secondly, in our experience large numbers of predicates are invented and there has to be someway of assessing which are the best to use for the absorption operator. The following algorithm can almost certainly be improved on, but even with this simple algorithm significant improvement in knowledge acquisition is achieved.

1. Start with an existing KB which may be empty.
2. Supply a case. If previous rules do not interpret this case correctly then goto step 3, otherwise repeat step 2 to add more cases.
3. Add some new rules to provide the correct conclusions for the case. (Note: we use a multiple classification system which provides multiple conclusions so that more than one may need to be corrected).
4. If the system did not interpret the case correctly because a generalized rule gave an incorrect conclusion, the invented predicate it used is moved from the heap of good predicates to a heap of bad predicates (see below). As well information about the generalized rule is stored to be used later in assessing the likely usefulness of new generalizations (See unlikely generalizations below)
5. Apply the intra-construction operator to each new rule added and any previous rule that gives the same conclusion as the new rule. The result is a set of new invented predicates. For simplicity a predicate consists of only two disjuncts. We ignore the possibility of constructing predicates with multiple disjuncts.
   - If any of new predicates are same as existing predicates in the good or bad heaps then they are deleted (see below).
6. Apply absorption
   - Apply absorption operator to each new invented predicate and all previous rules resulting in some generalized rules.
   - Apply absorption operator to each previous invented predicate (only from the good heap) and the new rule resulting in some generalized rules.
   - Delete any of the new generalized rules that have appear to be an unlikely generalization (see below)

7. Add the new predicates to the good heap. Test all predicates in the good heap against all the cases so far seen by the system. If the size of the heap is exceeded delete the predicates which cover the least number of cases. In this studies the size of the heap was set at 1000.
8. Store all the resulting rules in the KB.

Note that no rules, either as added by the expert or as generated or modified by intra-construction or generalized by absorption are removed from the KB. That is, although a an invented predicate may be moved from the good heap to the bad heap and not used for future generalization, it will remain in the knowledge base along with any previous rules that used it.
Unlikely generalizations

We use a very simple measure to assess if a newly generalized rule is likely to be of use. The basic idea is that if a generalization is found to cause an error, then any of the pairs of conditions from the original rule and the invented predicates might be at fault – except of course pairs of conditions that occur in the original rules. For example if we take parts of the previous example
The invented predicates were:
Z ← C, E (4)
Z ← A, F (5)
The rule
Y ← A, B, F (7)
was generalized to
Y ← B, Z (9)

If rule 9 now gives an incorrect conclusion, then the error might be due to any combination of conditions from the orignal rule (7) and the invented predicate rules (4 &5). The following is all possible pairs of conditions from these three rules, one condition from each rule, with pairs of conditions that occur in the original rule (7) being deleted.

AC, BC, FC, AE, BE, FE, ~~AA~~, ~~BA~~, ~~FA~~, ~~BF~~, ~~FF~~,

We maintain a list of all the ‘bad pairs’ that are seen over the life of the system. In step 6 above, we derive a similar list of pairs from each newly generalized rule. If more than a certain fraction of the rule’s pairs occur in the list of bad pairs, the new rule is deemed to be an unlikely generalization and deleted. In the studies a rule was deemed unlikely if more than 75% of its pairs occurred in the bad list.

4. Experimental evaluation

Evaluation of machine learning only requires appropriate data sets. However, evaluation of knowledge acquisition requires the availability of an expert. Experts rarely have the time available for this, particularly if a proper scientific study is proposed with control studies and comparison. To provide some sort of evaluation of knowledge acquisition we have previously developed a system of using a simulated expert, which can be put to work processing large numbers of cases under a variety of experimental conditions (Compton et al., 1994) (Kang, Compton & Preston, 1998). A simulated expert is a knowledge based system that is built by machine learning using one of the standard datasets. The ‘expertise’ it can provide to the RDR system that is being built is a classification of a case and some sort of rule trace indicating the features in the case that led to the conclusion. Of course this is a very limited source of knowledge and a long way from the capability of a real expert, but it seems to be the only low-cost way of getting some measure of the performance of a knowledge acquisition system.

We used J4.8 from the WEKA toolbench to generate a simulated experts from cases. We used three levels of expertise

1. P uses all the conditions from the rule trace. However, for this case we add some more refinement rules to cover every case in data set correctly.
2. G uses one less condition than in the rule trace (from P), selected randomly, unless this rule causes inconsistency to any seen cases.
3. S uses all the conditions from the rule trace (from P) but adds another .
4. W uses all the conditions from the rule trace J4.8 in default mode. This results in some errors in the data set. For this simulated expert, cases which have errors are removed from the data set before building the RDR KBS.

There is no particular merit in these different types of simulated expert; a range of 'expertise' is used simply to demonstrate that the results are not an artefact of using one particular expert. We used four data sets from the UC Irvine Data Repository: Car, Monk1, Dermatology and Nursery. Each simulation is run 5 times with the order of the data set randomized each time. We use the Mutiple classification (MCRDR) version of RDR (Kang & Compton, 1996)

1. 1.We start with a se t of cases.
2. We apply J4.8 to all the cases in a training set and develop a classifier able to classify all the cases in the training set. (See difference between the P and W protocols above)
3. Create an empty MCRDR knowledge based system..
4. Get one case randomly from collection of cases; input this case to MCRDR. Compare the classification from the MCRDR with the class of this labeled case. If the classification is correct repeat 4. If the classification is incorrect go to 5.
5. Pass the case to the simulated expert to provide a new rule. This rule is then passed to the learning generalization algorithm described above. (Although cases from the data sets have a single class, MCRDR may fire multiple rules and give multiple conclusions and the correction rule may have to be added to multiple places in the knowledge base)
6. Repeat 4

5. Results

The main results are shown in Table 1. In all cases there is a reduction in the number of rules the expert has to add. In the case of the Nursery and the Car data set the reduction in knowledge acquisition required is large. This is particularly to be expected for the Car data set which is a synthetic data set with a hierarchical structure (Zupan et al., 1997).
Table one. The number of rules added using generalisation compared to the number without generalisation. The different data sets and levels of expertise are show. The errors are the standard deviation of the 5 randomised studies for each data set and level of expertise

It should be noted that when generalization is used the expert also needs to add rules to correct errors from inappropriate generalization; despite this further requirement for rules to be added, there was still a decrease in the rules that had to be added. It can be also noted that different simulated expert policies do not effect the overall trend. We conclude from this that one might expect similar (or better) improvements when a genuine human expert is used.

Table one. The number of rules added using generalisation compared to the number without generalisation. The different data sets and levels of expertise are show. The errors are the standard deviation of the 5 randomised studies for each data set and level of expertise

DATA SET	WITH GENERALISATION	WITHOUT GENERALISATION	IMPROVEMENT
CAR P	49.4 ± 2.4	97.0 ± 0.0	49.0 %
CAR G	50.0 ± 3.7	87.0 ± 2.9	42.6 %
CAR S	127.8 ± 6.2	253.8 ± 7.1	49.6 %
CAR W	44.8 ± 3.0	70.0 ± 0.0	36.0 %
MONK1 P	14.0 ± 0.7	18.0 ± 0.0	22.0 %
MONK1 G	14.2 ± 1.9	19.2 ± 1.3	26.0 %
MONK1 S	33.2 ± 1.9	61.0 ± 4.7	46.0 %
MONK1 W	13.2 ± 1.1	18.0 ± 0.0	27.0 %
DERMATOLOGY P	49.8 ± 1.6	56.8 ± 1.6	13.4 %
DERMATOLOGY G	48.9 ± 1.6	56.6 ± 1.7	14.9 %
DERMATOLOGY S	72.4 ± 2.6	75.8 ± 4.4	4.5 %
DERMATOLOGY W	26.8 ± 1.1	28.0 ± 0.0	4.5 %
NURSERY P	43.2 ± 2.8	77.2 ± 1.3	44.0 %
NURSERY G	48.2 ± 1.5	56.6 ± 1.7	32.9 %
NURSERY S	69.8 ± 10.2	118.4 ± 5.1	41.0 %
NURSERY W	30.0 ± 3.0	47.0 ± 0.0	36.2 %

Figure 1, 2, 3 and 4 show the number of errors made against the total cases seen. The slope of this graph can be taken as a reasonable approximation to the error rate, as it is in essence the error rate on unseen samples. Note that these graphs do not show all the dataset; they are truncated at a maximum of a 1000 cases seen. The P expert only is shown.

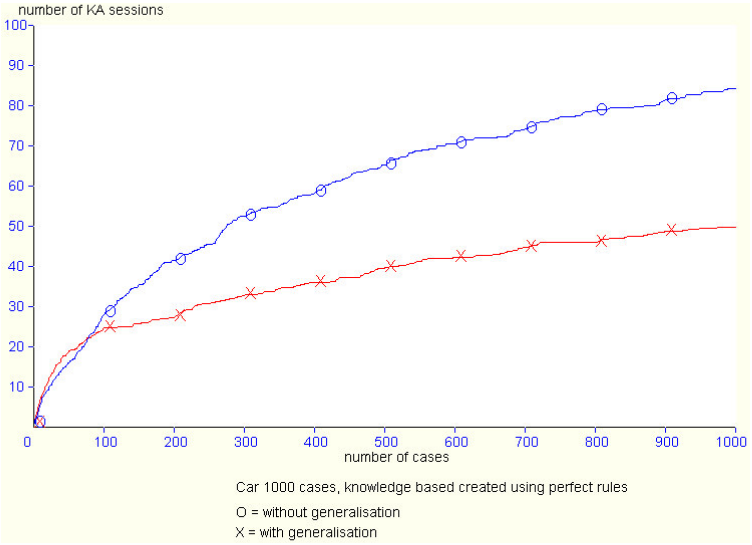

Figure 1. Error rate for Car data set

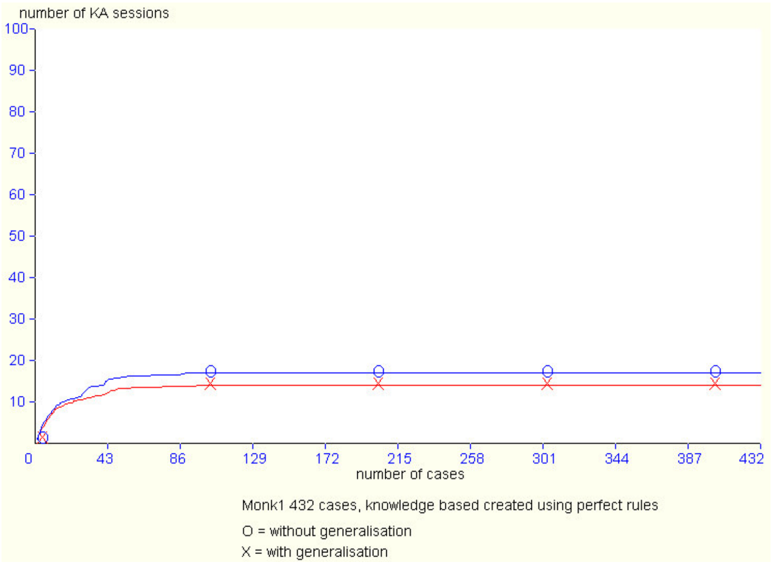

Figure 2. Error rate for Monk1 data set

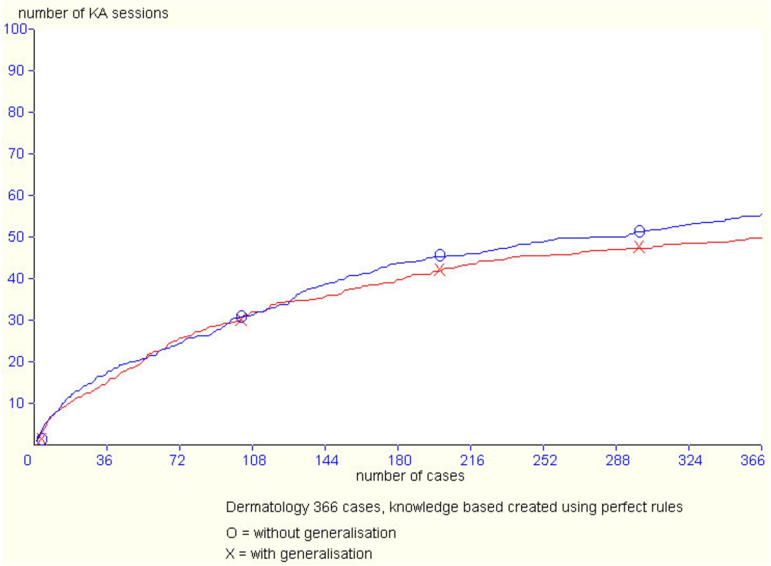

Figure 3. Error rate for Dermatology data set

As is the general case with RDR, the error rate of knowledge acquisition process rapidly plateaus. It plateaus to a similar error rate to good batch learning systems applied to the same domain. It can be noted that machine learning error rates are rarely zero and Catlett (1992) has observed that increasing the size of the training set generally changes the KB. The graphs here highlight that RDR allow the expert to keep on correcting errors if they wish.

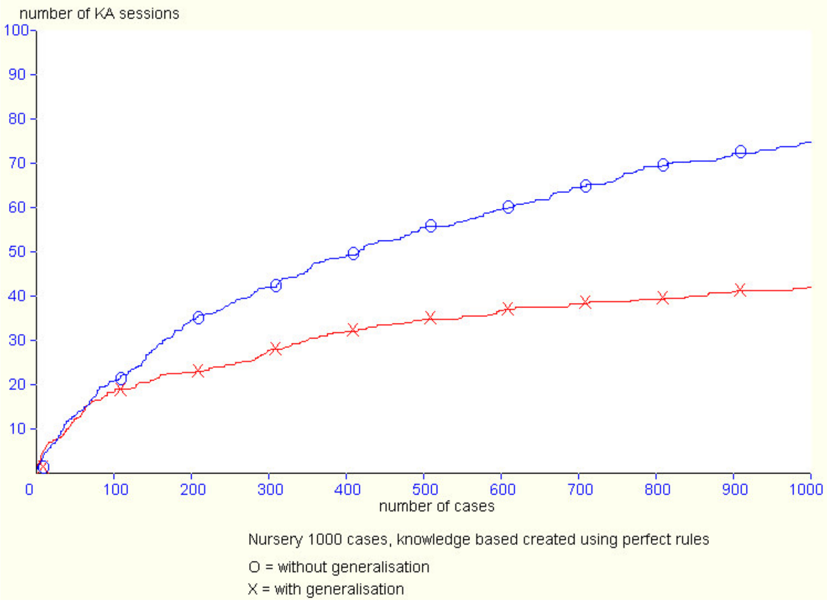

Figure 4. Error rate for Nursery dataset

For the Nursery and Car data set the curves plateau earlier with generalization, but there is essentially no difference for the other data sets.

6. Discussion

The system we have described is not proposed as a general machine learning system. It is specifically aimed at supporting knowledge acquisition from human experts. There is a clear and continuing need for knowledge acquisition from humans as there are many domains where suitable training data for machine learning is unavailable, and it is rare that systems developed by machine learning can be as subtle as a human experts in dealing with rare cases. On the other hand there are obviously many domains where massive datasets are readily available and there is little or no human expertise.

A learning system that supports knowledge acquisition should not increase the task of the human expert; it should leave the task of the expert as close as possible to the way they routinely function as an expert. The system we have developed fulfils this requirement and could apply in principle to many types of rule-based system; however, it fits particularly well with RDR. When combined with RDR the expert is only asked about conclusions should be made about the case, and what features in the data lead to that conclusion.

The decrease in the amount of knowledge acqusition required for two of the data sets suggests that it would be worthwhile testing the system human experts. Importantly the results also show that in the other two domains there is was not increase in the knowledge acquisition required. This suggests that use of the system will not degrade knowledge acquisition in domains where there are no useful intermediate concepts.

We spent a considerable amount of time trying to come up with ways of automatically managing inappropriate over-generalization, and selecting the best of the generalizations available. The heuristics we have eventually used are very simple and almost certainly better heuristics can be found. However, even with these heuristics, knowledge acquisition was reduced by close to 50%

in the car domain. One obvious improvement would be to replace a generalization by its original rule if it made an error on the first case it processed; at present we add a refinement rule regardless of whether the error is on the first or a later case. We will explore other heuristics in future research and also look at extending the approach to other operators beyond Boolean. Despite the practical usefulness of previous RDR systems based on a very flat structure, with rules directly linking primitive features and conclusion, these results confirm the value of heuristic classification with its intermediate structure – as suggested long ago by Clancey (1985). The difference here is that the discovery of intermediates is automated and hidden from the expert.

There has been a range of other work on integrating knowledge acquisition and machine learning (e.g. Tecuci & Kodratoff, 1995), but to our knowledge previous systems have not presented to the expert as a pure knowledge acquisition system with the machine learning hidden or have not been able to deal with data that was unclassified prior to the evolution of the knowledge base.

Acknowledgements

This research is partly funded by the Australian Research Council and by the Asian Development Bank. The authors would like to thank Rex Kwok for helpful discussions.

References

Beydoun, G. and A. Hoffman (1997). Acquisition of search knowledge. Knowledge Acqusition, Modeling and Management. E. Plaza and R. Benjamins. Berlin, Springer: 1-16.

Beydoun, G. and A. Hoffman (1998). Building problem solvers based on search control knowledge. Proceedings of the Eleventh Banff knowledge acqusition for knowledge-based systems workshop, B. Gaines and M. Musen Banff, SRDG Publications, University of Calgary, SHARE 3, 1-18.

Beydoun, G. and A. Hoffmann (2000). Incremental Acquisition of Search Knowledge. International Journal of Human Computer Studies 52(3): 493-530.

Bloedorn, E and Michalski R.S. (1991). Data Driven Constructive Induction in AQ17-PRE: A Method and Experiments, Proceedings of the Third International Conference on Tools for Artificiall Intelligence, San Jose, CA, November 9-14.

Catlett, J. (1992). Ripple Down Rules as a Mediating Representation in Interactive Induction. in Proceedings of the Second Japanese Knowledge Acquisition for Knowledge Based Systems Workshop. Kobe, Japan.

Clancey, W. J. (1985) Heuristic classification. Artificial Intelligence 27: 289-350.

Compton, P., R. Horn, R. Quinlan and L. Lazarus (1989). Maintaining an expert system. Applications of Expert Systems. J. R. Quinlan. London, Addison Wesley. 2: 366-385.

Compton, P. and R. Jansen.A. (1990). Philosophical basis for knowledge acquisition. Knowledge Acquisition, 1990. 2: p. 241--257.

Compton, P., P. Preston, B. Kang and T. Yip (1994). Local patching produces compact knowledge bases. A Future for Knowledge Acqusition: Proceedings of the European Knowledge Acquisition Workshop. L. Steels, G. Schreiber and W. Van de Velde. Berlin, Springer Verlag: 104-117.

Compton, P., Z. Ramadan, P. Preston, T. Le-Gia, V. Chellen and M. Mullholland (1998). A trade-off between domain knowledge and problem-solving method power. Proceedings of the Eleventh Banff knowledge acquisition for knowledge-based systems workshop, Banff, SRDG Publications, University of Calgary.

Compton P, Richards D, Generalising Ripple-Down Rules. (2000). Knowledge Engineering and Knowledge Management: Methods, Models, Tools, Eds. R. Dieng; O. Corby, Juan-les-Pins France, 2-6 Oct. Springer, Berlin, pp380-386.)

Drake, B. and G. Beydoun (2000). Predicate logic-based incremental knowledge acquisition. Proceedings of the sixth Pacific International Knowledge Acquisition Workshop, P. Compton, A. Hoffmann, H. Motoda and T. Yamaguchi Sydney, 71-88.

Edwards, G., P. Compton, R. Malor, A. Srinivasan and L. Lazarus (1993). PEIRS: a pathologist maintained expert system for the interpretation of chemical pathology reports. Pathology 25: 27-34.

Gaines, B.R. and P.J. Compton. (1992) Induction of Ripple Down Rules. in Proceedings of the Fifth Australian Conference on Artificial Intelligence. Hobart.

Kang, B., P. Compton and P. Preston (1995). Multiple Classification Ripple Down Rules : Evaluation and Possibilities. Proceedings of the Ninth Banff Knowledge Acquisition for Knowledge-Based Systems Workshop, B. Gaines and M. Musen Banff, Canada, University of Calgary, 17.1-17.20.

Kang, B., (1996) Validating Knowledge Acquisition: Multiple Classification Ripple Down Rules, PhD thesis in School of Computer Science and Engineering. 1996, New South Wales University: Sydney.

Kang, B. H., Yoshida, K., Motoda, H. and Compton, P. (1997) help desk system with intelligent interface. Applied Artificial Intelligence 11((7-8)): 611-631.

Kang, B., P. Compton and P. Preston (1998). Simulated Expert Evaluation of Multiple Classification Ripple Down Rules. Proceedings of the Eleventh Banff knowledge acqusition for knowledge-based systems workshop, B. Gaines and M. Musen Banff, SRDG Publications, University of Calgary, EVAL 4, 1-19.

Kivinen, J., H. Mannila, and E. Ukkonen. (1993). Learning Rules with Local Exceptions. in European Conference on Computational Theory.

Matheus, C. J., & Rendell, L. A. (1989). Constructive induction on decision trees. Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (pp. 645--650). Detroit, MI: Morgan Kaufmann.

Muggleton, S. (1987). Duce, An oracle-based approach to constructive induction. in Proceedings of the Tenth International Joint Conference on Artificial Intelligence.

Peter A. Flach and Nada Lavrac. (2000). The role of feature construction in inductive rule learning. Proceedings of the International Conference on Machine Learning 2000 workshop on Attribute-Value and Relational Learning: crossing the boundaries, Stanford, USA, Luc De Raedt and Stefan Kramer, pages 1 –11.

Richards, D. and P. Compton. (1997). Knowledge Acquisition First, Modelling Later. in Proceedings of the Tenth European Knowledge Acquisition Workshop.

Scheffer, T. (1996). Algebraic foundations and improved methods of induction or ripple-down rules. in Proceedings of the Second Pacific Rim Knowledge Acquisition Workshop.

Shiraz, G. and C. Sammut (1997). Combining Knowledge Acquisition and Machine Learning to Control Dynamic Systems. Proceedings of the fifteenth International Joint Conferences on Atrificial Intelligence. Nagoya, Japan, Morgan Kaufmann, 908-913.

Siromoney, A. and R. Siromoney, (1993). Variations and Local Exception in Inductive Logic Programming, in Machine Intelligence - Applied Machine Intelligence, S. Muggleton, Editor. p. 213 - 234.

Srinivasan, A. and King, R.D. (1996). Feature construction with inductive logic programming: A study of quantitative predictions of biological activity aided by structural attributes. In S. Muggleton, editor, Inductive Logic Programming: Proceedings of the 6th International Workshop. LNAI 1314, pp89--104. Springer.

Suryanto, H., D. Richards and P. Compton (1999). The Automatic Compression of Multiple Classification Ripple Down Rule Knowledge Base Systems: Preliminary Experiments. Proceedings of the Third International Conference on Knowledge-Based intelligent Information Engineering Systems. (IEEE Cat. No. 99TH8410), L. Jain Adelaide, pp203-206.

Suryanto H, Compton P. (2001) Discovery of Ontologies from Knowledge Bases, Proceedings of the First International Conference on Knowledge Capture, Eds. Yo-landa gil; Mark Musen; Jude Shavlik, Victoria, British Columbia Canada, 21-23 Oct. The Association for Computing Machinery, New York, USA, pp171-178.

Sutton, R. S., & Matheus, C. J. (1991). Learning polynomial functions by feature construction. Machine Learning: Proceedings of the Eighth International Workshop (pp. 208-212). Evanston, IL: Morgan Kaufmann.

Tecuci G. and Kodratoff Y. (Eds), (1995). Machine Learning and Knowledge Acquisition: Integrated Approaches , Academic Press.

Wada. T, Motoda. H, Washio. T. (2000). Integrating Inductive Learning and Knowl edge Acquisition in the Ripple Down Rules Method. Proceedings of the 6th Pa-cific Knowledge Acquisition Workshop, Sydney, Australia: p. 325-341.

Zupan, B., Bohanec, M., Demsar, J. and Bratko, I. (1997). Machine Learning by Function Decomposition. Proceedings of the 14 th International Conference on Machine Learning. (pp. 421-429). Morgan Kaufmann, San Francisco.