

Structured and Unstructured Induction with EDAGs

Brian R Gaines

Knowledge Science Institute
University of Calgary
Calgary, Alberta, Canada T2N 1N4
gaines@cpsc.ucalgary.ca, <http://ksi.cpsc.ucalgary.ca/KSI>

Abstract

Exception directed acyclic graphs (EDAGs) are knowledge structures that subsume trees and rules but can be substantially more compact. Manually constructed and induced EDAGs are compared by reconstructing Shapiro's "structured induction" of a chess end game. It is shown that the induced EDAG is very similar to that produced through consultation with experts, and that both are small, comprehensible solutions to a problem that is complex for people.

Introduction

A problem for knowledge discovery is that the trees or rules induced are not meaningful as "knowledge" (Quinlan 1991). Variant structures have been proposed that offer the possibility of more comprehensible models. Gaines' (1989) *Induct* induces rule graphs that cover common cases by default rules and infrequent cases through exception rules. Compton and Jansen's (1990) *ripple-down rules* generalize binary decision trees by allowing a node to contain a compound premise, and interior nodes to contain conclusions. Gaines (1991) shows that ripple-down rules are a particular case of *rules with exceptions* that can encode some knowledge structures more compactly. Oliver (1993) and Kohavi (1994) have shown how various forms of *decision graphs* may be induced and provide a more compact alternative than decision trees. Gaines (1995) generalizes these representations into *exception directed acyclic graphs* (EDAGs) that support exceptions within a general decision graph, and subsumes trees and rules.

However, none of the studies cited shows that the representation defined scales up to support the inductive modeling of large complex datasets in terms of a comprehensible knowledge structure. A problem in doing this is the lack of substantial datasets with well-defined expert models that allow inductive modeling to be humanly evaluated. Shapiro's (1987) study of a pawn versus rook end-game provides a complex dataset which has been modeled by a combination of human knowledge elicitation and inductive modeling. The main section of this article compares the model obtained direct induction of an EDAG for this problem with that obtained from human chess experts. The following introductory section illustrates the induction of EDAGs for a simple chess problem.

Modeling a Simple Chess Dataset

Quinlan (1979) gives ID3 models of 7 rook versus knight end game situations of increasing difficulty. The third problem involves 647 cases with 4 3-valued attributes, 3 2-valued attributes, and a 2-valued outcome. Figure 1 shows the decision tree induced by ID3 that solves this problem graphed as an EDAG in Induct. Figure 2 shows the C4.5 rules derived from this tree graphed as an EDAG.

Both EDAGs are used for decision making through the same procedure. The graph is traced from its root nodes. On any path, if the premise (if any) is true then the conclusion (if any) is noted for that path, replacing any previous conclusion noted for the path. A path is traced until a premise fails to hold, or a leaf node is reached. When a path has been traced, any conclusion noted is asserted.

This procedure is trivially applicable to decision trees and rules. It also applies to general graphs of rules with exceptions. Figure 3 shows the rules of Figure 2 with the conclusion "game = safe" taken as a default and common clauses in the premises factored by Induct. The numbers indicate that 613 cases are covered by the default, and 34 are exceptions.

Figure 4 shows a solution with multi-level exceptions induced by Induct. The numbers indicate that the exceptions may themselves be represented as a default covering 30 cases, an exception to that covering 10 cases, and an exception to that covering 4 cases.

The EDAGs of Figures 3 and 4 are logically equivalent to those of Figures 1 and 2. However, it would be tedious to explain the decision tree or rules in words whereas the small EDAG in Figure 3 may be explicated as: "the game is safe unless the black king, rook and knight are in line, the rook bears on the king, the rook bears on the knight, and either the king to knight distance is 3, or the knight to white king distance is 1, and either the king to rook distance is 2 or 3, or the rook to white king distance is 1."

The graphic representation in the EDAGs seems more perspicuous than the text. EDAGs retain the flow of decision making that makes decision trees attractive. In addition, the reduced restrictions of the directed acyclic graph structure enables them to avoid some of the unnecessarily complex structures generated by redundant replication of nodes in a decision tree representing a disjunctive structure.

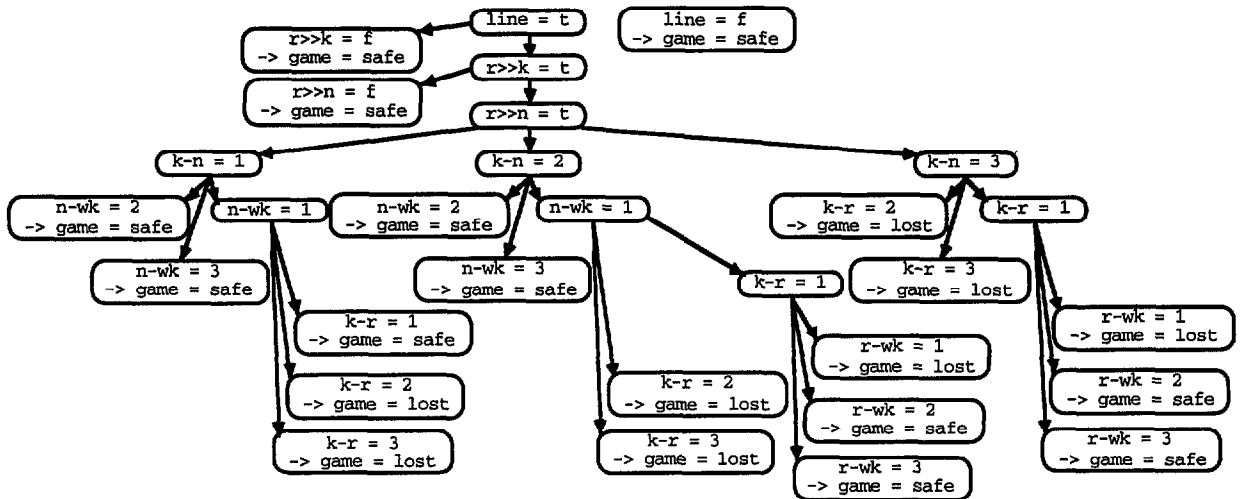


Figure 1 ID3 decision tree solving a rook versus knight chess end game problem

line = f -> game = safe	r >> k = f -> game = safe	r >> n = f -> game = safe
k-n = 1 n-wk = 2 -> game = safe	k-n = 2 n-wk = 2 -> game = safe	k-r = 1 r-wk = 2 -> game = safe
k-n = 1 n-wk = 3 -> game = safe	k-n = 2 n-wk = 3 -> game = safe	k-r = 1 r-wk = 3 -> game = safe
line = t r >> k = t r >> n = t k-r = 2 n-wk = 1 -> game = lost	line = t r >> k = t r >> n = t k-r = 3 n-wk = 1 -> game = lost	line = t r >> k = t r >> n = t n-wk = 1 r-wk = 1 -> game = lost
line = t r >> k = t r >> n = t k-n = 3 r-wk = 1 -> game = lost	line = t r >> k = t r >> n = t k-r = 2 k-n = 3 -> game = lost	line = t r >> k = t r >> n = t k-r = 3 k-n = 3 -> game = lost

Figure 2 C4.5 rules derived from the decision tree of Figure 1

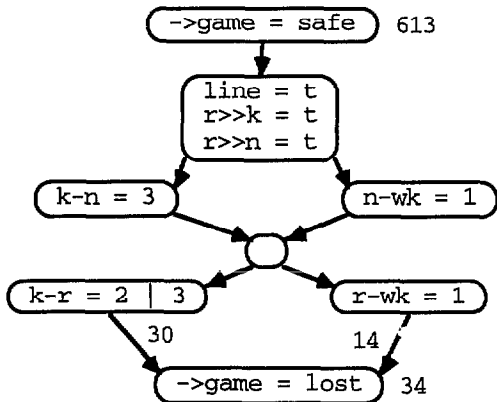


Figure 3 Induct factorization of C4.5 rules

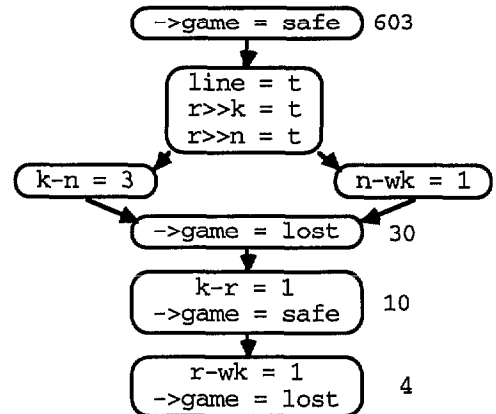


Figure 4 Induct rules with exceptions

Comparing Elicited and Induced Structures

How do the results of the previous section scale up to a more complex situation? Shapiro (1987) studied a situation in the rook versus pawn (KPa7KR) end-game situation which he notes is poorly documented in the chess literature. The white pawn is at a7, able to queen, and the black rook and the two kings are in any of the 209,718 legal configurations of which 129,825 are classified won-for-white and 79,893 are classified not-won-for-white.

In the chess situations studied by Quinlan and Shapiro the board positions of the pieces are well-defined but the attributes that a chess expert uses in assessing those positions are not. Hence, human experts were interviewed to elicit those characteristics of the situations that they saw as significant to making a decision about whether the game was won or not. Quinlan's studies elicited these attributes, applied them to a set of situations where the outcome was known, and then used ID3 to develop a decision tree which would be totally correct if the attributes adequately characterized the problem.

Shapiro's study addressed the problem noted in the introduction, that the resultant decision trees, although correct, could not be regarded as demonstrating "knowledge" of chess. He had chess experts partition the problem situation and define small sub-problems, each of which defined a new decision variable. He defined "small" to mean that the sub-problem could be solved using fewer than 7 attributes, each of which could be derived from a board position with less than a "screenfull of C code." For each sub-problem, he classified the database of cases in terms of the small set of attributes and the decision variable, and then used ID3 to develop a decision tree.

Shapiro termed this technique "structured induction" and used it to develop a complete solution of the KPa7KR end-game in terms of 9 sub-problems introducing 8 additional decision variables. The total number of attributes defined is 35 binary and 1 ternary, and the total number of different attribute vectors involved is 3,196. This dataset is available through the University of Irvine archive (Murphy and Aha 1994), and has been used in other studies (Muggleton 1987; Holte, Acker and Porter 1989).

Structured Induction using Decision Trees

Figure 5 is reproduced from Shapiro's book and shows the structure of his solution in terms of the attributes used to develop the 9 sub-trees. Each circle represents a decision tree with between 4 and 7 attributes, and the complete structure has 122 nodes.

Figure 6 shows the top-level decision tree, pa7 in Figure 5. It involves the introduction of intermediate decision variables, such as ds (good delayed skewer threat) and dq (simple delay to queening), which are computed through other decision trees.

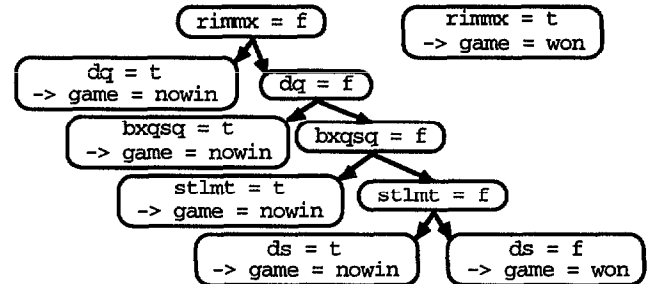


Figure 6 Top-level tree pa7 (Shapiro 1987)

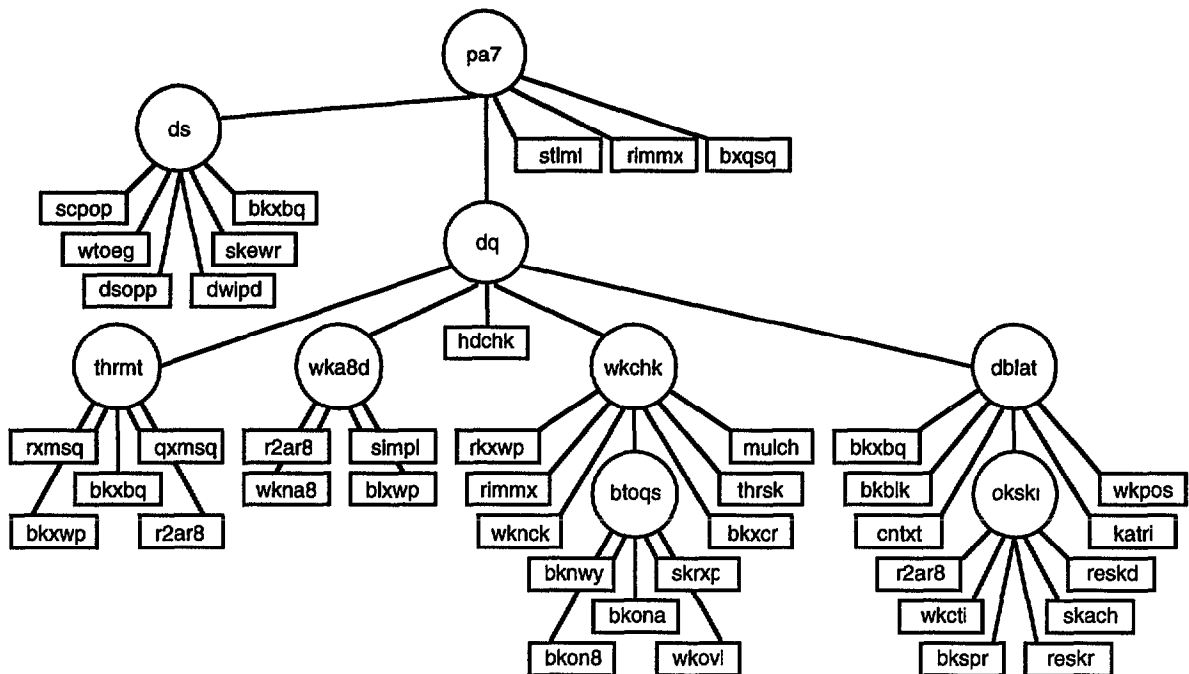


Figure 5 Structured induction of solution to a chess end game (Shapiro 1987)

Structured Induction using EDAGs

Shapiro's use of decision trees reflects the representational schema available to him in the early 1980s. The 9 sub-problem trees are themselves fairly difficult to understand, and are incomprehensible when combined into a single knowledge structure. As a first experiment ID3 was replaced with Induct and an EDAG induced for each sub-problem as shown in Figure 7. Enclosures have been added to show the sub-problems and their relationships. A nested EDAG computes an intermediate decision variable used by the EDAG at the level of nesting above.

It can be seen at the bottom right that the game is won unless the black rook cannot be captured safely ($rmmx = f$) and either there is a simple delay to queening ($dq = t$) or one or more black pieces controls the queening square ($bxqsq = t$), or the white king is in stalemate ($stlmt = t$), or there is a good delayed skewer threat ($ds = t$). It can be seen at the bottom left that there is a good delayed skewer threat ($ds = t$) if there is a special opposition pattern ($spcop = t$), or a combination of 5 other attributes have the values specified. Each of the 8 intermediate decision variables is determined by a relatively simple and comprehensible EDAG. Those for $wka8d$ (white king on promotion square) and $wkchk$ (white king in check) are interesting because Induct has generated two-level exceptions for them.

Thus, the reconstruction of Shapiro's results in terms of EDAG's confirms more clearly than did the original ID3

trees that the problem has been structured into simple sub-problems. The sub-problems are also meaningful because they correspond to situations defined by the expert in terms comprehensible to a chess player.

Figure 8 shows the EDAGs of Figure 7 converted to a single EDAG by elimination of the intermediate decision variables. This is simple if the EDAG computing the variable has a single exception which is the truth value specified when the variable is used. There is one sub-problem, $okskr$, where the exception is single but the exception is the opposite to that required. The variable is eliminated by treating $okskr = f$ as an exception in the EDAG for $dblatt$. This involves adding two additional clauses ($mulch = f$, $bkxcr = f$) which correspond to alternative paths to nowin that the simple exception does not take into account. Similarly, the elimination of the $wkna8$ and $wkncck$ variables computed through multi-level exceptions requires the introduction of the additional variables marked by bullets.

Figure 8 is an EDAG developed through structured induction that solves the KP7KR problem. The numbers on the graph show the numbers of cases out of the total of 3196 that are covered by each part of the EDAG. It can be seen that the top level default accounts for 1616 cases, the path through $scop = t$ for 1 case, and so on. The numbers decided through all paths total more than 3196 because a single case may flow through more than one path.

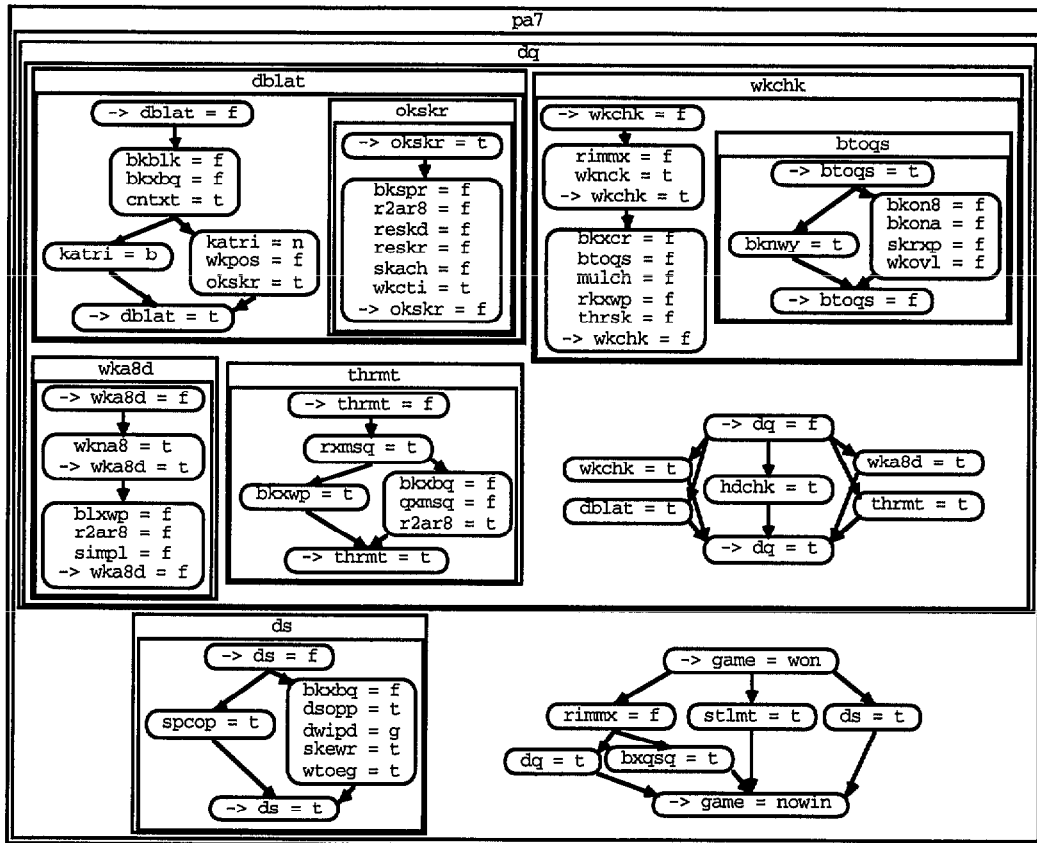


Figure 7 EDAGs induced for each of Shapiro's 9 sub-problems

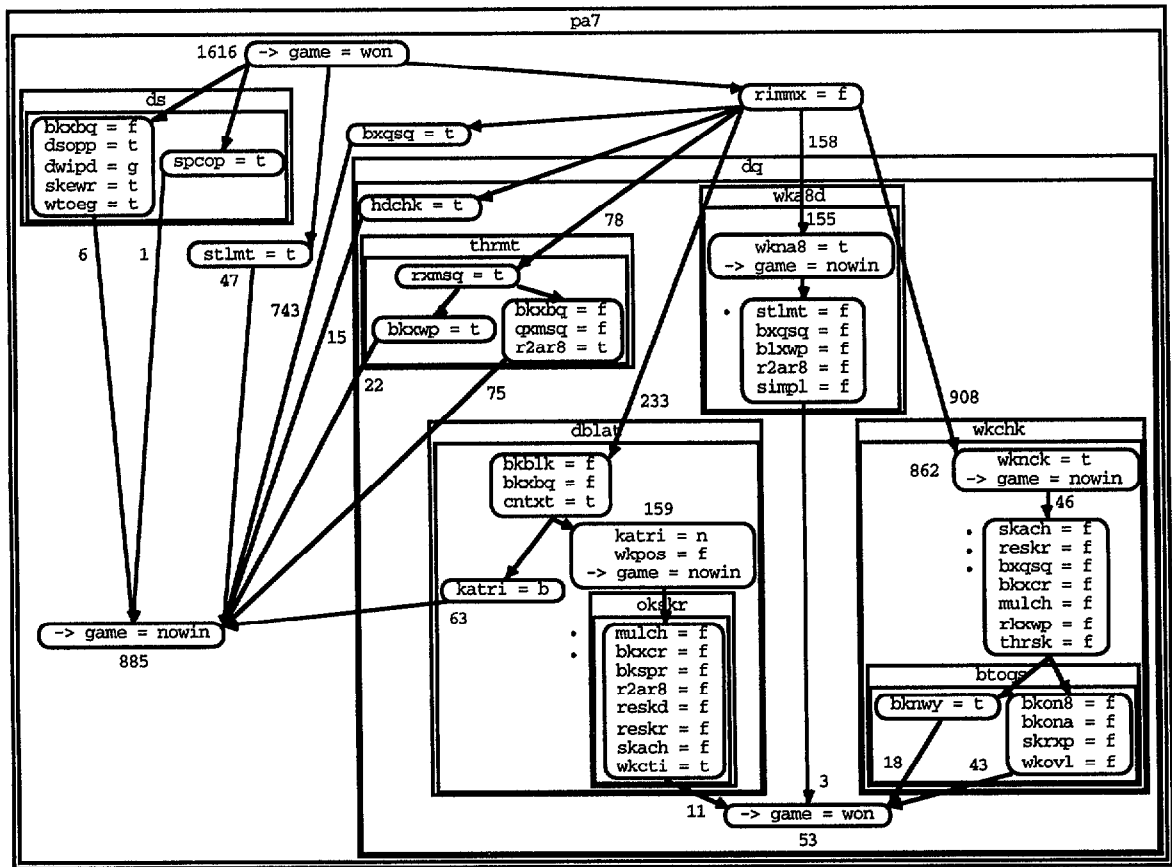


Figure 8 EDAG obtained from Figure 7 by eliminating the intermediate variables

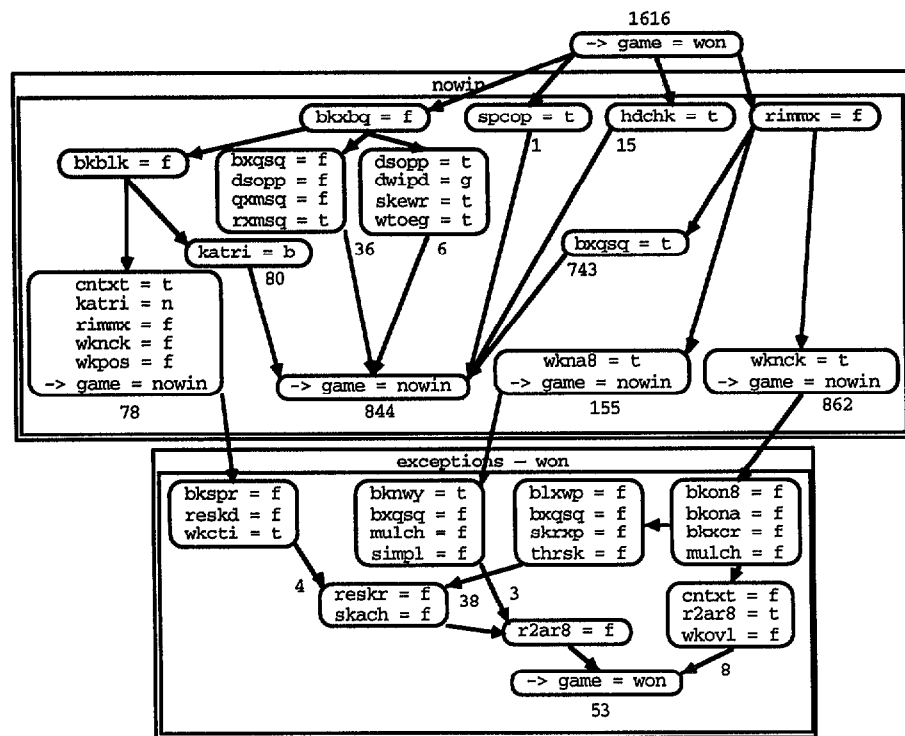


Figure 9 EDAG induced directly by Induct

Comparing a Directly Induced Solution with that of Structured Induction

The EDAG of Figure 8 is a relatively simple exposition of the solution to a difficult chess end-game situation, and it exemplifies the induction of "knowledge" as "a correct and effectively computable descriptions that can be assimilated and used by a human being" (Quinlan 1991). However, it was developed with a mixture of sub-problem structuring by human experts and inductive modeling. It is interesting to investigate whether a similar solution can be developed through inductive modeling without human structuring.

Figure 9 shows the solution induced by Induct directly from the dataset of 3,196 cases. One difference from the EDAG of Figure 8 is that three of the attributes, bkwpx, rkwpx and stlmt, have not been used. Of these only stlmt plays a major role in Figure 8 accounting for 47 cases. However, if this path is removed only 2 errors occur as 45 of the cases are also covered by other paths. These 2 cases are covered by the rule $wkna8 = t \rightarrow nowin$ in Figure 9.

Apart from the path through $stlmt = t$, it can be seen the top level structure above the exceptions in Figure 9 largely reproduces that of Figure 8. One difference is that some structures such as $hdchk$ and $dblatt$ are under $rwxm = f$ in Figure 8 and not in Figure 9. It turns out that it makes no difference to insert $rwxm = f$ in these paths in Figure 9. It is a redundant clause that Induct eliminates. Another difference is that the additional clause, $wknc = f$, has been introduced into the compound clause in ds . This corresponds to considering the delayed skewer threat only when the white king is not in check which is reasonable. It seems likely that the chess experts would have intended this when defining the skewer situation, but it was not taken into account when generating the cases for ID3.

The 53 exceptions to the top level structure are not generated through the same structures in Figures 8 and 9. The original $okskr$ exception in $dblatt$ is reproduced exactly, but the exceptions relating to $wkna8$ and $wknc$ are only loosely related in the two EDAGs. However, these fairly rare and complex exception situations were simplified artificially in the Shapiro study using the ≤ 7 attributes and ≤ 1 screen of C criteria. They do not correspond to basic chess concepts as do the upper level attributes.

Conclusions

Research on producing inductive models that correspond to those of human experts is made difficult by the lack of test case data where expert models are available. Shapiro's work provides an excellent opportunity to investigate the induction of comprehensible knowledge structures. The similarity of the solutions induced directly from the dataset and generated from experts shows the role of the expert in construing a problem. Even though the expert was absent in the development of the model of Figure 9, his or her knowledge was still present in the attributes used. The "background knowledge" of how to perceive a game was adequate for Induct to learn to solve the problem.

In conclusion, this study indicates that the EDAG knowledge structure that has previously been shown to subsume trees and decision rules while allowing more concise representations does scale up to a significant larger problem. It also shows that the knowledge structure produced by automatic induction of an EDAG can be very similar to that elicited directly from an expert. Many similar studies are necessary to substantiate these conclusions, but the lack of datasets with expert associated expert models will make such studies slow to occur. Meanwhile EDAGs are of interest as attractive alternatives to decision trees and rules.

References

- Compton, P. and Jansen, R. 1990. A philosophical basis for knowledge acquisition. *Knowledge Acquisition* 2(3) 241-258.
- Gaines, B.R. 1989. An ounce of knowledge is worth a ton of data: quantitative studies of the trade-off between expertise and data based on statistically well-founded empirical induction. *Proceedings of the Sixth International Workshop on Machine Learning*. pp.156-159. San Mateo, California: Morgan Kaufmann.
- Gaines, B.R. 1991. Integrating rules in term subsumption knowledge representation servers. *AAAI'91: Proc. Ninth National Conference on Artificial Intelligence*. pp.458-463. Menlo Park, California: AAAI Press/MIT Press.
- Gaines, B.R. 1995. Transforming rules and trees into comprehensible knowledge structures. *Knowledge Discovery in Databases II*. AAAI/MIT Press.
- Holte, R.C., Acker, L.E. and Porter, B.W. 1989. Concept learning and the problem of small disjuncts. *IJCAI'89: Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*. pp.813-818. San Mateo: Morgan Kaufmann.
- Kohavi, R. 1994. Bottom-up induction of oblivious read-only decision graphs: strengths and limitations. *AAAI'94: Proceedings of the Twelfth National Conference on Artificial Intelligence*. pp.613-618. Menlo Park, California: AAAI Press/MIT Press.
- Muggleton, S. 1987. Structuring knowledge by asking questions. *Progress in Machine Learning*. pp.218-229. Wlmslow, UK: Sigma Press.
- Murphy, P.M. and Aha, D.W. 1994. UCI Repository of machine learning databases. Department of Information and Computer Science, University of California at Irvine.
- Oliver, J.J. 1993. Decision graphs - an extension of decision trees. *Proceedings of 4th International Workshop on AI and Statistics*. pp.343-350.
- Quinlan, J.R. 1979. Discovering rules by induction from large collections of examples. *Expert Systems in the Micro Electronic Age*. pp.168-201. Edinburgh: Edinburgh University Press.
- Quinlan, J.R. 1991. Foreword. *Knowledge Discovery in Databases*. pp.ix-xii. Cambridge, Mass.: MIT Press.
- Shapiro, A.D. 1987. *Structured Induction in Expert Systems*. Wokingham, UK: Addison-Wesley.