Proc. of the ONR Workshop on Knowledge Acquisition, Crystal City, VA, Nov. 5-7, 1989.

KNOWLEDGE BASE REFINEMENT AS IMPROVING AN INCORRECT, INCONSISTENT AND INCOMPLETE DOMAIN THEORY[1]

David C. Wilkins and Kok-Wah Tan
Department of Computer Science
University of Illinois
405 North Mathews
Urbana, IL 61801

ABSTRACT

We view automation of knowledge base refinement as improvements to a domain theory. Techniques are described that we have developed to handle three types of domain theory pathologies: incorrectness, inconsistency, and incompleteness. The major sources of power of our learning method are a confirmation theory that connects the domain theory to underlying domain theories, the use of an explicit representation of the strategy knowledge for a generic problem class (e.g., heuristic classification) that is separate from the domain theory (e.g., medicine) to be improved, and lastly an explicit, modular, and declarative knowledge representation for the domain theory.

INTRODUCTION

One central problem of expert systems is knowledge base (KB) refinement (Buchanan and Shortliffe, 1984). The major research efforts that have directly confronted this problem include TEIRESIAS (Davis, 1982), INDUCE (Michalski, 1983), ID3 (Quinlan, 1983), SEEK (Politakis, 1984), AQUINAS (Boose, 1984), RL (Fu, 1985), MORE (Kahn, 1985), and ODYSSEUS (Wilkins, 1987). Historically, these efforts evidence an evolution along three important dimensions: automation of the refinement process, refinement of more complex representations, and greater diversity in the types of refinements.

There has been considerable progress in automation of the three principal subtasks of KB refinement, which are deficiency detection, suggestion of a repair, and validation of a repair. TEIRESIAS, the first refinement program, facilitated these three subtasks for EMYCIN-based expert systems (Davis, 1982), by providing an intelligent editor that allowed a cooperating human expert to accomplish *manually* the subtasks. In contrast, when ODYSSEUS refines a HERACLES-based expert system (HERACLES is a descendant of EMYCIN), the three refinement subtasks are automated (Wilkins, 1987). Evolution in refinement of more complex representations is seen in systems which are able to repair many types of rule (e.g., heuristic, definitional) and frame knowledge (e.g., subsumption, hierarchical, causal slots). Evolution in diversity of repairs is seen in systems where many types of domain theory pathologies are correctable.

This paper describes an integrated set of methods that have been developed to correct automatically three types of KB pathologies: incorrectness, inconsistency, and incompleteness. We begin with necessary background on the architecture and method of knowledge representation of the expert system to be improved. The methods for handling each of the three KB pathologies are then described.

ProHC HEURISTIC CLASSIFICATION SHELL

Our learning methods are designed to improve any knowledge base crafted for the ProHC expert system shell (Tan, 1989). ProHC is a refinement of HERACLES, based on the experience gained in creating the ODYSSEUS apprenticeship learning program for HERACLES (Wilkins, 1987). HERACLES is itself a refinement of EMYCIN, based on the experience gained in creating the GUIDON case-based tutoring program for EMYCIN (Clancey,

[1]This paper was presented at the Sixth International Workshop on Machine Learning, Cornell University, June 29-July1, 1989.

1986). These shells use a problem-solving method called *heuristic classification*, which is the process of selecting a solution out of a pre-enumerated solution set, using heuristic techniques (Clancey, 1985). The primary application KB for ProHC and HERACLES is the NEOMYCIN medical KB for diagnosis of meningitis and similar neurological disorders (Clancey, 1984). This section describes the types of knowledge encoded in ProHC and HERACLES, and how ProHC differs from HERACLES.

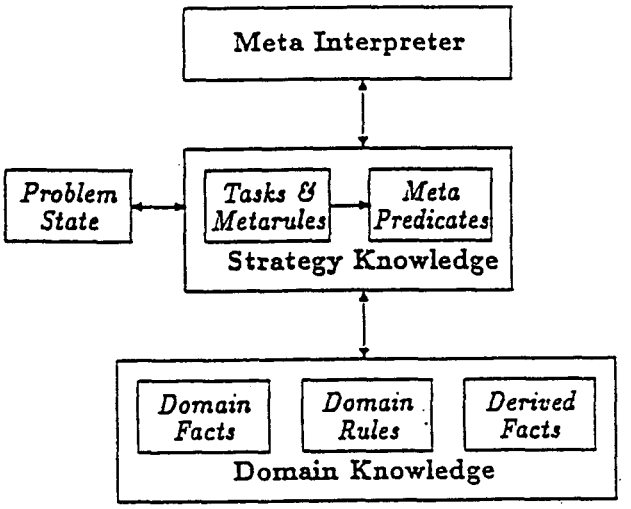

Figure 1: ProHC System Architecture.

Domain knowledge consists of Mycin-like rules and simple frame knowledge for an application domain (e.g., medicine, geology). An example of rule knowledge in Horn clause format is conclude(migraine-headache, yes, .5) :- finding(photophobia, yes), meaning 'to conclude the patient has a migraine headache with a certainty .5, determine if the patient has photophobia'. An example of frame knowledge is subsumed-by(viral-meningitis, meningitis), meaning 'hypothesis viral meningitis is subsumed by the hypothesis meningitis.' *Problem state knowledge* is generated during execution of the expert system. Examples of program state knowledge are rule-applied(rule163), which says that Rule 163 has been applied during this consultation, and differential(migraine-headache, tension-headache), which says that the expert system's active hypotheses are migraine headache and tension headache.

Strategy knowledge is contained in the shell, and approximates a cognitive model of heuristic classification problem solving. The different problem-solving strategies that can be employed during problem solving are explicitly represented, which facilitates use of the model to follow the the line-of-reasoning of a human problem solver. The strategy knowledge determines what domain knowledge is relevant at any given time, and what additional information is needed to solve a the problem. The problem state and domain knowledge, including rules, are represented as tuples. Strategy metarules are quantified over the tuples.

Some of the ways in which ProHC differs from HERACLES at the meta-level are as follows. First, the task interpreter consists of logic metainterpreter that uses a blackboard agenda mechanism to decide which task and metarule to execute next. Second, metarule premises do not change the state of the system, do not call other tasks, do not have procedural attachment to LISP code, and do not call more than one subgoal in their action part. Third, all meta-level control state information, such as task end condition flags, have been eliminated. The major difference between HERACLES and ProHC at the domain level is the use of Pearl's method to represent rule uncertainty and for propagating information in a hierarchy of diagnostic hypotheses (Pearl, 1986). The differences facilitate solving the learning global credit assignment problem (comparing the behavior of an expert to the expert system and noticing when there is a significant difference)

and the learning local credit assignment problem (determining the specific knowledge differences between the expert and the expert system).

INCORRECT DOMAIN THEORY

The KB refinement process described in this paper assumes an initial faulty KB, created, for example, by interviewing experts. Refinement goes through three stages that address the problems of *incorrect*, *inconsistent*, and *incomplete* domain theory knowledge, respectively.

The method used to handle incorrect domain knowledge relies on a *confirmation theory* and an *underlying domain theory*. A confirmation theory is a decision procedure that when given an arbitrary candidate tuple of domain knowledge, can decide whether that tuple is true or false; it connects the tuples in the domain theory to the underlying domain theory. An underlying domain theory consists of knowledge that can underpin the knowledge in the domain theory. This allows validation of the initial KB for all types of knowledge for which a confirmation theory exists. Tuples that do not pass the test are deleted or modified. The confirmation theory frequently changes the strength of heuristic rules supplied by experts by recomputing their strength based on a case library.

INCONSISTENT DOMAIN THEORY

The KB may contain knowledge tuples that are individually correct, but that interact deleteriously with other pieces of knowledge during problem solving, and thus give an inconsistent domain theory. A major source of inconsistency for classification expert systems involves heuristic rules that are *collinear variants*. Such rules fire on almost exactly the same cases. Higher-order correlation information is usually not available, so collinear variants cause hypotheses to have incorrect strengths. Our confirmation theory for heuristic rules detects and removes collinear variants.

Another source of inconsistency arises if a KB is sociopathic (Wilkins and Ma, 1989). By definition, this occurs when the individual rules are good rules, but there exists a subset of the rule set that gives better performance than the entire rule set. Correcting this problem has been proved to be NP-hard. We use a heuristic method, called the sociopathic reduction algorithm, to reduce sociopathicity.

INCOMPLETE DOMAIN THEORY

We have developed two methods for extending an incomplete domain theory: an apprenticeship learning approach, and a case-based reasoning approach. Table 1 shows the major refinement steps and the method of achieving them for apprenticeship and case-based learning. The techniques will be elaborated below.

Apprenticeship Learning Approach

Apprenticeship learning is a form of learning by watching, in which learning occurs as a byproduct of building explanations of human problem-solving actions. An apprenticeship is the most powerful method that human experts use to refine and debug their expertise in knowledge-intensive domains such as medicine. The major accomplishment of our method of apprenticeship learning is showing how an explicit representation of the strategy knowledge for a general problem class, such as diagnosis, can provide a basis for learning the knowledge that is specific to a particular domain, such as medicine.

Apprenticeship learning involves the construction of explanations, but it is different from explanation based learning as formulated in EBG (Mitchell et al., 1986) and EBL (DeJong, 1986). It is also different from explanation based learning in LEAP (Mitchell et al., 1985), even though LEAP also focuses on the problem of improving a knowledge-based expert system. In EBG, EBL, and LEAP, the domain theory is capable of explaining a training instance and learning occurs by generalizing an explanation of the training instance. In contrast, in our apprenticeship research, a learning opportunity occurs when the domain theory, which

Learning Method	*Case-Based Learning* (similarity-based)	*Apprenticeship Learning* (explanation-based)
Scope	Heuristic rules.	Heuristic rules. 4 other types of relations.
Detect KB deficiency	Select and run a case. Deficiency exists if case is misdiagnosed.	Observe expert solving a case. Deficiency exists if action of expert cannot be explained.
Suggest KB deficiency	Generalize or specialize rules. Induce new rules.	Find tuples that allows explanation to be completed under single fault assumption.
Validate KB repair	Use underlying domain theory to validate repairs.	Use underlying domain theory to validate repairs.

Table 1: Comparison of case-based and apprenticeship learning method for extending an incomplete domain theory.

is the domain KB, is incapable of producing an explanation of a training instance. The domain theory is incomplete or erroneous, and all learning occurs by making an improvement to this domain theory.

The first stage of learning involves the detection of a KB deficiency. Explanations are are constructed for each of an expert's observed problem-solving actions. When ODYSSEUS observes the expert asking a "findout" question, such as asking if the patient has visual problems, it finds all explanations for this action. When none can be found, an explanation failure occurs. This failure suggests that there is a difference between the knowledge of the expert and the expert system and it provides a learning opportunity. ODYSSEUS assumes that deficient domain knowledge is the cause of the explanation failure.

The second step is to conjecture a KB repair. The confirmation theory can judge whether an arbitrary tuple of domain knowledge is erroneous, independent of the other knowledge in the KB (Wilkins, 1987). Hence, when a KB deficiency is detected during apprenticeship learning, we assume the problem is missing knowledge. The search for the missing knowledge begins with the single fault assumption. The missing knowledge is conceptually a single fault, but because of the way the knowledge is encoded, we can learn more than one tuple when we learn rule knowledge. Conceptually, the missing knowledge could be eventually identified by adding a random domain knowledge tuple to the KB and seeing whether an explanation of the expert's findout request can be constructed. How can a promising piece of such knowledge be effectively found? Our approach is to apply backward chaining to the findout question metarule, trying to construct a proof that explains why it was asked. When the proof fails, it is because a tuple of domain or problem state knowledge needed for the proof is not in the knowledge base. If the proof fails because of problem state knowledge, we look for a different proof of the findout question. If the proof fails because of a missing piece of domain knowledge, we temporarily add this tuple to the domain knowledge base. If the proof then goes through, the temporary piece of knowledge is our conjecture of how to refine the knowledge base.

The third step is to evaluate the proposed repair. To do this, we use a confirmation theory containing a decision procedure for each type of domain knowledge that tells us whether a given tuple is acceptable. The current confirmation theory provides an underpinning for 5 of 19 domain tuple types. The confirmation theory for heuristic rules uses a case library, and uses a set of biases for judging rule quality.

Case-Based Learning Approach

The case-based learning approach currently modifies or adds heuristic rules to the KB. It runs all the cases in the library and locates those that are misdiagnosed. Given a misdiagnosed case, the local credit assignment problem is solved as follows. The premises of the rules that concluded the wrong final diagnosis are weakened

by specialization, and the premises of the rules that concluded the correct diagnosis are strengthened. If this does not solve the problem, new rules will be induced from the patient case library that apply to the misdiagnosed case and that conclude the correct final diagnosis. The verification procedure used to test all KB modifications is identical to that described for apprenticeship learning.

EXPERIMENTAL RESULTS

Some preliminary testing has been completed, expanding on results reported earlier (Wilkins, 1988). These tests used the NEOMYCIN KB for neurological disorders (constructed manually by interviewing experts over many years), and a collection of 114 solved cases that were obtained from records at the Stanford Medical Hospital. Table 2 shows the various diseases and their sample sizes in the evaluation set. The result of each test suite are described along three dimensions. TP (true positive) refers to the number of cases that the expert system correctly diagnosed as present, FN (false negative) to the number of times a disease was not diagnosed as present but was indeed present, and FP (false positive) to the number of times a disease was incorrectly diagnosed as present.

ProHC with the manually constructed NEOMYCIN KB diagnosed 32 of the 112 cases correctly (28.5% accuracy).

Disease	*Number* Cases	KB1			KB2			KB3			KB4		
		TP	FN	FP	TP	FN	FP	TP	FN	FP	TP	FN	FP
Bacterial Meningitis	16	14	2	49	14	2	21	12	4	4	14	2	13
Brain Abscess	7	0	7	1	0	7	1	5	2	15	1	6	0
Cluster Headache	10	1	9	0	7	3	4	7	3	4	8	2	0
Fungal Meningitis	8	0	8	0	4	4	0	3	5	0	3	5	0
Migraine	10	4	6	6	1	9	0	4	6	0	6	4	0
Myco-TB Meningitis	4	0	4	2	4	0	0	4	0	0	4	0	1
Primary Brain Tumor	16	0	16	0	0	16	0	0	16	0	3	13	0
Subarach Hemorrhage	21	1	20	0	15	6	0	16	5	2	16	5	3
Tension Headache	9	7	2	5	7	2	6	7	2	6	8	1	3
Viral Meningitis	11	5	6	11	10	1	12	10	1	6	10	1	12
None	0	0	0	6	0	0	6	0	0	7	0	0	7
Totals	112	32	80	80	62	50	50	68	44	44	73	39	39

Table 2: Summary of ProHC experiments. The KB1 column is the performance using the manually constructed domain theory. KB2 shows performance after use of methods that correct an incorrect domain theory. KB3 and KB4 show the performance after using case-based learning and apprenticeship learning, respectively, to extend the incomplete domain theory.

The first stage of improvement involves locating and modifying *incorrect* domain knowledge tuples. Our method modifies 48% of the heuristic rules in the KB. The improvement obtained using the refined KB is shown in column KB2 of Table 2; ProHC diagnosed 62 cases correctly (55.3% accuracy), showing an improvement of about 27%. The second stage of improvement involves correcting *inconsistent* domain knowledge. No experimental results are reported here, although our methods have been previously shown to lead to significant improvement (Wilkins and Ma, 1989).

The third stage of improvement involves extending a correct but incomplete domain KB. Two experiments were conducted. The first used case-based learning. All the cases were run, and two misdiagnosed cases in areas where the KB was weak were selected. The case-based learning approach was applied to these two cases. This refinement, shown in column KB3 of Table 2, enabled the system to diagnose 68 cases correctly

(60.7% accuracy), showing an aggregate improvement of 32%. The second experiment used apprenticeship learning. The experimental setup involved watching a physician diagnosing two cases not in the set of 112. This refinement, shown in column KB4 of Table 2, enabled the system to diagnose 73 cases correctly (65.2% accuracy), an aggregate improvement of about 37%.

More experimental work remains. Our previous experiments with ODYSSEUS suggest that the apprenticeship learning approach is better than a case-based approach for producing a use-independent KB to support multiple problem-solving goals such as learning, teaching, problem-solving and explanation generation.

SUMMARY AND CONCLUSIONS

The long-term objectives of this research are the creation of learning methods that can harness an explicit representation of generic shell knowledge and that can lead to the creation of use-independent KB that rests on deep underlying domain models. Within this framework, this paper describes specialized methods that address three major types of KB pathologies: incorrect, inconsistent, and incomplete domain knowledge. We believe that the use of *specialized methods* for different domain knowledge pathologies, and an *ordered sequential correction* as described in this paper will minimize the interactions between pathologies and thereby make the problem much more tractable.

Acknowledgements

We would like to express our deep gratitude to Lawrence Chachere, Ziad Najem, Young-Tack Park for their major role in the design and implementation of the ProHC shell and for many fruitful discussions. This research was supported by ONR grant N00014-88K0124 and an Arnold O. Beckman research award from the University of Illinois. Marianne Winslett provided valuable comments on draft versions of this paper.

References

Buchanan, B. G. and Shortliffe, E. H. (1984). *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Reading, Mass.: Addison-Wesley.

Clancey, W. J. (1984). NEOMYCIN: Reconfiguring a rule-based system with application to teaching. In Clancey, W. J. and Shortliffe, E. H., editors, *Readings in Medical Artificial Intelligence*, chapter 15, pages 361–381. Reading, Mass.: Addison-Wesley.

Clancey, W. J. (1986). From GUIDON to NEOMYCIN to HERACLES in twenty short lessons. *AI Magazine*, 7:40–60.

Davis, R. (1982). Application of meta level knowledge in the construction, maintenance and use of large knowledge bases. In Davis, R. and Lenat, D. B., editors, *Knowledge-Based Systems in Artificial Intelligence*, pages 229–490. New York: McGraw-Hill.

DeJong, G. (1986). An approach to learning from observation. In Michalski, R. S., Carbonell, J. G., and Mitchell, T. M., editors, *Machine Learning, Volume II*, volume 2, chapter 19, pages 571–590. Los Altos: Morgan Kaufmann.

Mitchell, T. M., Keller, R. M., and Kedar-Cabelli, S. T. (1986). Explanation-based generalization: A unifying view. *Machine Learning*, 1(1):47–80.

Mitchell, T. M., Mahadevan, S., and Steinberg, L. I. (1985). LEAP: A learning apprentice for VLSI design. In *Proceedings of the 1985 IJCAI*, pages 573–580, Los Angeles, CA.

Pearl, J. (1986). On evidential reasoning in a hierarchy of hypotheses. *Artificial Intelligence*, 28:9–15.

Tan, K. (1989). Knowledge base validation and refinement for a heuristic classification expert system. Master's thesis, University of Illinois at Urbana-Champaign.

Wilkins, D. C. (1987). *Apprenticeship Learning Techniques For Knowledge Based Systems.* PhD thesis, University of Michigan. Also, Knowledge Systems Lab Report KSL-88-14, Dept. of Computer Science, Stanford University, 1988, 153pp.

Wilkins, D. C. (1988). Knowledge base refinement using apprenticeship learning techniques. In *Proceedings of the 1988 National Conference on Artificial Intelligence*, pages 646–651, Minneapolis, MN.

Wilkins, D. C. and Ma, Y. (1989). Sociopathic knowledge bases. Technical Report UIUCDCS-R-89-1538, Department of Computer Science, University of Illinois. submitted to *Artificial Intelligence.*