ELSEVIER

Knowledge-Based Systems 9 (1996) 253–266

Knowledge-Based SYSTEMS

Integrating machine learning with knowledge acquisition through direct interaction with domain experts

Geoffrey I. Webb

School of Computing and Mathematics, Deakin University, Geelong, Victoria, 3217, Australia

Received 12 July 1995; revised 11 December 1995; accepted 18 January 1996

Abstract

Knowledge elicitation from experts and empirical machine learning are two distinct approaches to knowledge acquisition with differing and mutually complementary capabilities. Learning apprentices have provided environments in which a knowledge engineer may collaborate with a machine learning system allowing for a synergy between the complementary approaches. The Knowledge Factory is a knowledge acquisition environment that allows a domain expert to collaborate directly with a machine learning system without the need for assistance from a knowledge engineer. This requires a different form of environment to the learning apprentice. The paper describes techniques for supporting such interactions and their implementation in a knowledge acquisition environment called The Knowledge Factory.

Keywords: Knowledge acquisition; Machine learning; Apprenticeship learning

1. Introduction

Although data-driven machine learning is often viewed as an autonomous process, it is increasingly apparent that the application of machine learning to real world problems requires substantial input from a domain expert before satisfactory solutions can be derived. Indeed, real world applications of machine learning often involve a cycle of interaction in which a knowledge engineer, in consultation with a domain expert,

- defines a search space (the attributes and values or predicates and atoms to be considered) and configures other parameters to the learning algorithm,
- applies the algorithm,
- reviews the proposed solutions,
- iterates through a process of refining the definition of the search space and manipulating the learning system's other parameters guided by the results obtained by the learning algorithm and insight provided by the domain expert.

At the very least, before machine learning can begin, a domain expert must be consulted in order to determine the primitive terms (attributes, predicates, etc.) with which the machine learning system is to work. Thus, the successful application of machine learning requires application of knowledge elicitation from experts.

However, knowledge elicitation from experts is subject to many constraints. Experts are poor at articulating executable models that capture their expertise [1]. Further, there exist domains that cannot be tackled by current human expertise but for which there is promise of the derivation of expertise by machine learning [2].

Each of these approaches to knowledge-base development has distinct and complementary capabilities. Given this, there is considerable potential for gain through collaboration between a machine learning system and a human expert during knowledge acquisition.

A number of systems called learning apprentices have been developed to this end [3–11]. These systems are all distinguished by the need for sophisticated knowledge engineers.

By contrast, The Knowledge Factory is such a system that has been designed for direct use by domain experts. This alternative design objective has led to a substantially different type of system to the learning apprentice. It has required the use of a knowledge representation formalism that is readily understood and manipulated by non-knowledge-engineers and has required the development

PII S0950-7051(96)01033-7

of sophisticated communication mechanisms that enable the human and computer partners to communicate diverse forms of knowledge in a readily mastered manner. This system is intended for knowledge acquisition tasks for which

- the tasks can be handled by a domain expert without aid from an expert knowledge engineer,
- a domain expert has nontrivial insight,
- it is possible to accumulate a suitable set of example cases.

The resulting approach to knowledge acquisition falls between standard machine learning, learning apprentices and automated knowledge elicitation aids such as ripple-down rules [12] and automated knowledge acquisition systems [13–16]. It is distinguished from standard machine learning by the tight integration of the domain expert into the knowledge acquisition process. It differs from learning apprentices by removing the knowledge engineer from the process. It is distinguished from the ripple-down rules and repertory grid approaches to automated knowledge elicitation by the removal of the reliance on the domain expert's ability to articulate appropriate rules for the domain.

2. Relative capabilities of machine learning and knowledge acquisition from experts

Before considering how machine learning and knowledge acquisition from experts should be integrated, it is illuminating to consider strengths and weaknesses of each approach. Human experts are able to draw upon book and orally transmitted learning (the accumulated wisdom of thousands of years of human endeavor), practical experience, general and common-sense knowledge. They are situated in the social and operational context in which the knowledge is to be employed. A machine learning system is able to perform exhaustive logical and/or statistical analysis of large sets of examples and may be free from conceptual bias derived from book learning and social context.

Thus, a machine learning system may

- consider solutions which a human expert's biases lead him or her to overlook,
- infer knowledge from examples in contexts in which human expertise is unable to provide insight, for example because no one has previously developed solutions to the problem under consideration

The first of these points is multifaceted. The types of biases that human experts are likely to have, and which may be desirable to overcome, include not only social biases, such as racism and sexism, but also methodological biases. Methodological biases can be introduced by training (for example where the expert is told to attend to specific factors only) or by practice (the expert becomes accustomed to considering specific factors only). In contrast, the machine learning system will employ such biases only if explicitly directed to do so.

Conversely, human expertise may

- provide solutions in circumstances for which the available example cases are not adequate for a machine learning system to be able to derive a suitable solution,
- discriminate between potential solutions on the basis of knowledge external to the formal considerations available to the machine learning system, for example where, although a solution is correct, it will not be acceptable to some of the intended users.

Further, a machine learning system is only able to operate within the constraints of the metamodel with which it is provided. This metamodel is the set of primitive knowledge representation terms, predicates and operators with which the system is provided. If a particular factor is necessary to produce an accurate model of a domain and the machine learning system is not provided with knowledge of that factor, it is not possible for the machine learning system to produce an accurate model. The metamodel defines the space of possible models that the machine learning system can generate. If an adequate model does not lie within that space, then it is not possible for the machine learning system to find one. While constructive induction [17–31] may construct new complex terms from given primitive terms, it is unable to construct new terms that can only be described using terms that lie outside the given vocabulary.

By contrast, human experts are accomplished at extending and refining metamodels as circumstances demand and they have ready access to multiple sources of insight that may help guide such a process. Examples of such sources of insight include

- skills and experience with problem formalisation and description,
- extensive domain specific and general knowledge and experience,
- access to other knowledge sources, such as books and other experts, as required.

It seems clear from this simple survey of comparative capabilities that each of these two approaches to knowledge acquisition has the potential to complement the other.

3. Knowledge representation scheme

Sophisticated knowledge representation schemes, such as first-order logic, are simply not appropriate for use by non-knowledge-engineers. It requires considerable training and experience to master such formalisms. Rather, a simple attribute-value production rule formalism has been adopted. The antecedents are conjunctions of simple tests on attributes (set membership for categorical attributes and ranges of values for ordinal attributes). The consequents are simple classification statements.

In the current version of the system, the rule sets are flat. The consequent of one rule cannot be the antecedent of another. While some liberalisation of this restriction might be considered, it would be inappropriate to allow complex chaining to any great depth, as non-knowledge-engineers are unlikely to readily master the complexities of such a rule base. Attempts to incorporate more sophisticated knowledge representation devices, such as model construction operators [32], are also viewed as likely to impose an undesirable barrier to use for those without extensive training in knowledge engineering.

Example cases are described by a simple vector of attribute values.

We have conducted informal experimentation with two sets of experts: medical practitioners and financial analysts. The former have had little computing expertise and the latter have had substantial computing expertise but no background in knowledge engineering. This experimentation has shown that these experts take readily to this form of knowledge representation. They have no difficulty in interpreting existing rules, or specifying new rules or examples.

4. Machine learning capabilities

Whereas most research into machine learning has examined techniques for creating new knowledge bases, any nontrivial collaboration for knowledge acquisition will require that the machine learning system be able to refine successive drafts of the knowledge base. Such refinement should be able to occur after and take full account of any form of input that the human expert may provide.

Such a refinement system should be able to consider a wide range of different types of refinement to a knowledge base, but should restrict the degree of change that the system may wreak upon the existing knowledge base, or at least those sections that the expert has proposed. It will in most cases be unacceptable to the user to have the machine learning system appear to disregard previous input to the knowledge base. Examples of such refinement techniques include those that are designed to identify a single refinement at a time [4, 9] and those that are designed to modify an entire rule base (or set of rules leading to a single conclusion) at a time [5, 10, 33, 34].

Each of these approaches is likely to be suited to different machine–expert collaborative knowledge-acquisition contexts. Indeed, it may be desirable for a general purpose system to support both modes of operation. The induction of a complete rule base will, in most circumstances, be desirable if the induction system is developing the first draft of the knowledge base (if the initial knowledge base is empty). In other contexts, selection between the induction of a single refinement or a complete set of refinements at a time is most likely to be a matter of personal preference.

A machine learning system has far more to offer than the simple revision of a rule set, however. A machine learning system is capable of analyzing the quality of a rule set and of identifying specific deficiencies. Machine learning systems provide as standard mechanisms for applying a set of rules to an evaluation set of example cases and summarising the resulting performance. Further analysis can be provided by examining the performance of the rules on the training cases. Mechanisms for communicating such analysis to the user will add greatly to the opportunity for synergistic interaction between the system and the user.

5. Human expert capabilities

The power and flexibility of the facilities under the human partner's control are as important as the power and flexibility of the machine learning component. Whenever the human partner feels that he or she has something to contribute it is valuable to have a mechanism by which to make that contribution.

At the very least it is essential that the human partner be able to make arbitrary changes to the knowledge base. The direct ability to edit the knowledge base is simple to provide. All that is required is an appropriate knowledge base editor. However, as important as the immediate capacity for the user to perform editing is that the machine learning component should both note and respect during induction the changes previously made by the expert. This is why it is essential that the machine learning system should attempt to minimise the change that it wreaks upon the knowledge base.

The Knowledge Factory achieves this through the use of the DLGref2 machine learning algorithm [34]. This algorithm is an extension of the DLG learning algorithm [35] that supports knowledge base refinement. It explicitly seeks to minimise the extent to which the initial knowledge base is transformed during induction.

Also important is the ability of the user to provide general advice to the machine learning system, such as placing constraints upon the rules that can be developed. Of similar importance is the ability to identify deficiencies in the knowledge base without being required to specify a solution. Mechanisms to these ends are discussed below in the section on communication.

It is desirable that the expert be able to specify both precise rules (rules which he or she believes to be correct) and approximate rules (rules which he or she believes capture aspects of a correct solution but which may require further refinement). The Knowledge Factory supports both forms of user supplied rules through the mechanism of rule annotations. Each rule in the knowledge base is annotated with a qualitative evaluation. If the user indicates that a rule has high quality, then the machine learning component will not modify it. This is the appropriate action for precise rules. If the user indicates that a rule has intermediate quality, the machine learning component is free to modify it. This is the appropriate action for approximate rules.

This mechanism could be further extended by allowing annotation of individual clauses within the condition of a rule. Thus the user could specify that some clauses should not be altered but that others may. However, it is not clear that the extra facility that this would introduce warrants the added complexity that it would entail, in terms of both representing to the user the status of small subparts of the knowledge base, and making explicit the exact consequence of each such specification. Due to these considerations, such a facility has not, at this stage, been implemented in The Knowledge Factory.

It is also important that the human partner be able to modify and extend the metamodel within which the knowledge base is constructed. That is, it should not only be possible to modify rules, but also possible to create new primitive terms and new predicates and to alter the ranges of existing predicates. For example, if the initial metamodel defines temperature as a predicate over the domain low, medium and high, the domain expert should be able to change the range of the predicate to a real number.

While this point may appear trivial, it is frequently the case that a domain expert will realise that a particular metamodel is inadequate only after considerable knowledge-base development has occurred. Indeed, it may only be possible to determine that a particular metamodel is inadequate by first attempting to create a knowledge base within it.

It is important that changes to the metamodel should be supported in such a manner as to enable as much as possible of the knowledge base developed under the old metamodel to be transferred into the new model. It will greatly hinder progress if it is necessary to start afresh every time that a new metamodel is required.

Although there has been considerable research into machine learning techniques for extending metamodels by learning new terms and predicates (constructive induction) the new terms and predicates so learned are necessarily derived from the primitive terms and predicates that are supplied externally. Such approaches extend the space of possible solutions but still fail to enable a machine learning system to develop adequate solutions when the initial metamodel lacks key terms, predicates or operators. Morik [36] has investigated techniques for identifying deficient metamodels but still requires knowledge elicitation to rectify the identified deficiencies.

In view of these factors, it seems apparent that

- metamodel revision is an important knowledge-base development capability,
- this capability can only be satisfactorily provided by a human expert; machine learning cannot provide it on its own.

The Knowledge Factory allows the user to modify the metamodel by adding or deleting new attributes at any stage. When a new attribute is added, all existing example cases are extended to incorporate the new attribute with the value set to 'unknown'. When an attribute is deleted, it is immediately deleted from all example cases and rules.

The Knowledge Factory's model revision facility could usefully be extended to enable the type of an attribute to be altered. For example, categorical valued attributes could be converted to have ordinal values, or the number of values in a categorical attribute could be increased or decreased, with appropriate adjustment of all existing rules and example cases.

The capacity for metamodel revision is one of the more important extensions that collaboration with an expert provides to autonomous machine learning.

6. Communication

Successful collaboration requires the ability for the collaborating parties to communicate [37]. Communication requires a language. It is not feasible with current technology to support domain independent specialist natural language interaction. It is also possible that such communication may be undesirable, even if it were feasible, as other communications mechanisms may prove more effective. One alternative is the use of a formal language for communication. However, the need for the human expert to learn a formal language of sufficient power and complexity to support sophisticated discussion about a knowledge base and its strengths and deficiencies would present a tremendous barrier to the integration of machine learning with knowledge elicitation from experts. The choice of a suitable language or languages for communication is going to be vital to the success or failure of a collaborative system.

The aim of knowledge acquisition is to produce a knowledge base. This knowledge base must be expressed within a formal language, the target language. This target language could be used for communication between the collaborating parties. However, while it is important to be able to communicate in the target language, it is also important to be able to communicate about expressions in the target language. As many knowledge representation languages do not support metastatements, this will often require the use of an alternate language for communication. As already stated, the use of a complex formal language for this purpose creates a major barrier to the use of such a system.

6.1. Case-based communication

The consideration of example cases provides an alternative paradigm for communication to that of formal knowledge-representation languages. Experts are accustomed to and proficient at considering cases and communicating about expertise through the consideration of cases. Further, it is precisely the analysis of example cases that machine learning systems are structured around and good at performing. In consequence, consideration of example cases provides a powerful mechanism for communication between the human expert and the induction system. A number of systems have used example cases to communicate between a machine learning system and an expert, notably MARVIN [38]. The following discussion seeks to define and extend these techniques.

6.2. Machine initiated case-based communication

Cases form a natural means of communicating the reasons for the machine learning system's actions to a human user. The primary consideration on which a machine learning system bases its decisions is how well the rules perform on example cases. Thus, one of the key aspects of a correct answer to a question about why a particular rule was developed by a machine learning system is that it correctly handled a particular set of cases. Further, the key to why a particular rule was not developed will usually be that it fails to correctly handle a particular set of cases. While it is true that this is not the complete explanation (other factors, such as the relative complexity of the two rules, may also be involved), it is clear that the use of example cases provides a powerful medium for accurately communicating to the user key aspects of the underlying basis for the induction system's decisions.

However, communication through example cases is not limited to a coarse explication of whether an individual case is handled correctly or incorrectly. Cases can be analyzed in two dimensions. In one dimension they are distinguished by whether a rule has fired or not. In the other dimension they are distinguished by whether it would be appropriate for that rule to fire or not. Cases for which a rule fires and for which it is appropriate for that rule to fire (covered positive examples) provide direct evidence of the value of the rule. Similarly, cases for which the rule has not fired and for which it is not appropriate for it to fire (uncovered negative examples) also provide evidence of the rule's value. Cases for which the rule has fired and for which it is not appropriate for it to fire (covered negative examples) provide evidence against the rule. Similarly, cases for which the rule has failed to fire, but for which it would be appropriate to fire also provide evidence of potential shortcomings of the rule.

Covered examples provide evidence relating to potential specialisations of a rule. Covered positive examples can be used to explore the limits to potential specialisations to a rule. It will usually be undesirable to specialise a rule so that it ceases to cover a covered positive example.

By contrast, covered negative examples provide guidance for the potential extent for specialisations. It will generally be desirable to specialise a rule so that it no longer covers a covered negative example.

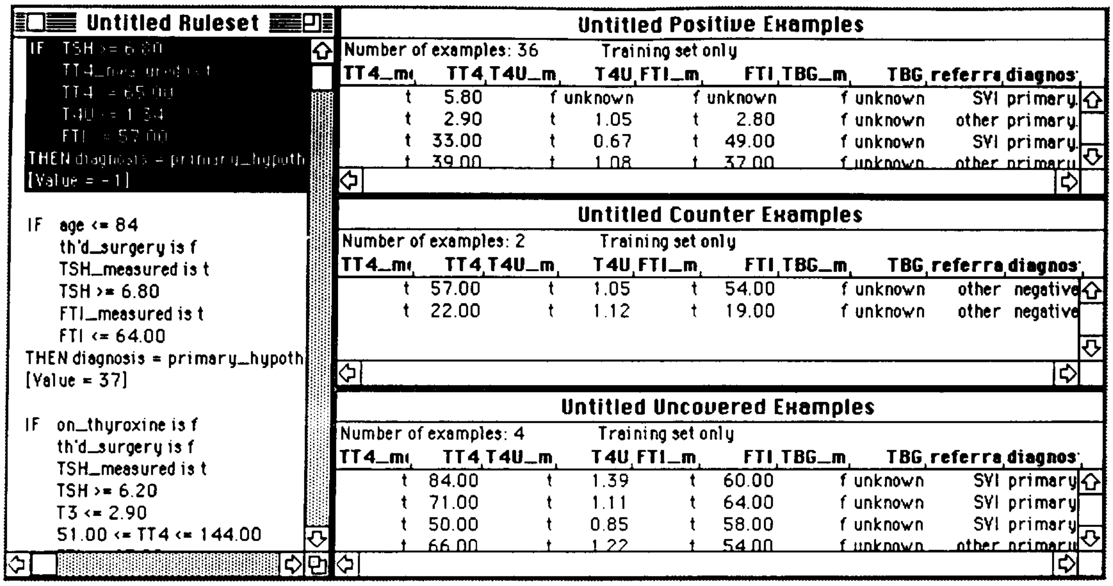
Similarly, uncovered examples provide evidence relating to potential generalisations. Uncovered positive examples can be used to find ways in which it might be useful to generalise a rule whereas uncovered negative examples place constraints on the desirable generalisations.

Each of these classes of examples can be used by the machine learning component to communicate different information to the human expert. If asked why a rule was developed it can reply by showing the rule's covered positive examples and uncovered negative examples. If asked why an alternative rule was not developed it can reply by showing the covered negative examples and uncovered positive examples for that rule. In general, the four sets of examples provide a general summary of the machine learning system's evaluation of any rule or potential rule.

In addition to evaluation of individual rules, case-based critique can also be used to evaluate a rule set as a whole. Of interest are the cases that the rule set handles correctly, those for which it produces inappropriate conclusions and those for which it fails to produce a conclusion.

Rather than providing a mechanism for the human expert to pose questions for which the different sets of examples can be used as answers, The Knowledge Factory at all times provides an explicit list of each of the covered positive, the covered negative and the uncovered positive examples of the current rule and of the mishandled and not handled examples for the rule set as a whole. These lists are of such high value and are so frequently consulted that it would become a burden to require the user to explicitly request them each time that they are to be consulted. Uncovered negative examples are not provided as it is natural for the human user to expect negative examples to not be covered and thus it is only important to draw attention to violations of this expectation in the form of covered negative examples.

In addition to consideration as to whether individual



Untitled Ruleset

```
IF   TSH >= 6.80
     TT4_measured is t
     TT4 <= 65.00
     T4U <= 1.34
     FTI <= 57.00
THEN diagnosis = primary_hypoth
[Value = -1]

IF   age <= 84
     th'd_surgery is f
     TSH_measured is t
     TSH >= 6.80
     FTI_measured is t
     FTI <= 64.00
THEN diagnosis = primary_hypoth
[Value = 37]

IF   on_thyroxine is f
     th'd_surgery is f
     TSH_measured is t
     TSH >= 6.20
     T3 <= 2.90
     51.00 <= TT4 <= 144.00
```

Untitled Positive Examples

Number of examples: 36 Training set only

TT4_m	TT4	T4U_m	T4U	FTI_m	FTI	TBG_m	TBG	referra	diagnos
t	5.80	f	unknown	f	unknown	f	unknown	SVI	primary
t	2.90	t	1.05	t	2.80	f	unknown	other	primary
t	33.00	t	0.67	t	49.00	f	unknown	SVI	primary
t	39.00	t	1.08	t	37.00	f	unknown	other	primary

Untitled Counter Examples

Number of examples: 2 Training set only

TT4_m	TT4	T4U_m	T4U	FTI_m	FTI	TBG_m	TBG	referra	diagnos
t	57.00	t	1.05	t	54.00	f	unknown	other	negative
t	22.00	t	1.12	t	19.00	f	unknown	other	negative

Untitled Uncovered Examples

Number of examples: 4 Training set only

TT4_m	TT4	T4U_m	T4U	FTI_m	FTI	TBG_m	TBG	referra	diagnos
t	84.00	t	1.39	t	60.00	f	unknown	SVI	primary
t	71.00	t	1.11	t	64.00	f	unknown	SVI	primary
t	50.00	t	0.85	t	58.00	f	unknown	SVI	primary
t	66.00	t	1.22	t	54.00	f	unknown	other	primary

Fig. 1. Individual rule and related example displays.

rules handle cases correctly or incorrectly, it is important to consider whether a rule set as a whole handles cases correctly. Each individual rule in a rule set might appear to handle a particular case appropriately while the outcome from applying the rule set is inappropriate, due to interactions between rules. To this end it is also valuable to have lists of the example cases that a rule set as a whole fails to handle appropriately.

The availability of each of these lists of examples also serves the important role of training set verification. It is not unknown for errors to be included in the set of example cases that are made available to the machine learning system. These will often cause the induction of anomalous rules. Investigation of these rules, through perusal of the appropriate lists of examples, is likely to lead to the identification of the erroneous examples which can then be corrected. The Knowledge Factory allows a case to be edited in place in any window in which it is displayed.

During its operation, The Knowledge Factory continually displays the relevant lists of example cases, with each list in a separate window. The primary window presents the current rule set. At all times, one of these rules is distinguished as 'the current rule'.

The user has available at most times windows displaying

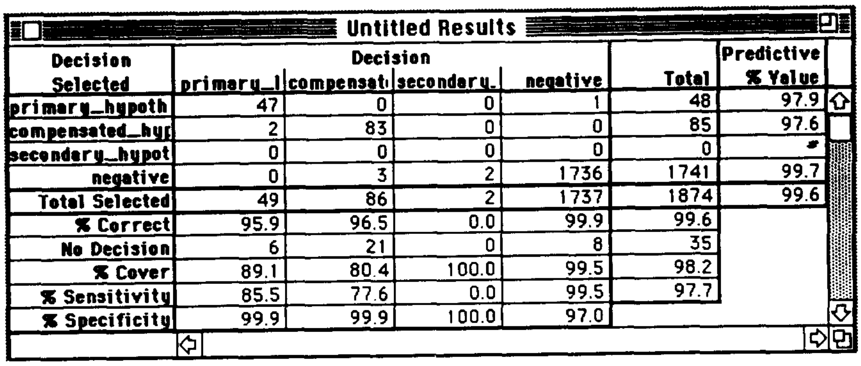
- the current knowledge base,
- all examples,
- all covered positive examples for the current rule,
- all counter (covered negative) examples for the current ruler,
- all uncovered positive examples for the current rule,
- all examples for which it is not possible to determine (due to missing values) whether or not they are covered by the current rule,
- all examples for which the rule set as a whole reaches the wrong conclusion,
- all examples for which the rule set as a whole reaches no conclusion.

Fig. 1 illustrates how the system displays a rule and the relevant sets of examples that relate to the rule. The sets of examples that relate to the performance of the rule set as a whole are also displayed in the same format. This simple mechanism has proved straightforward for use by non-knowledge-engineers.

6.3. Summary rule set evaluation

Another use for example cases is to provide a summary of the general accuracy of a rule set when applied to a set of cases. Such summaries are routinely used to evaluate the performance of machine learning systems. However, these evaluations are usually restricted to the evaluation of predictive accuracy. This is far from the only useful measure of rule quality [39]. Indeed, raw predictive accuracy will often be relatively unimportant as a measure of the quality of an expert system. Consider, for example, a medical diagnostic system in a context for which there is a high cost for false negatives (undiagnosed patients will suffer serious consequences) and a low cost for false positives (patients falsely diagnosed as suffering from the disease undergo a treatment that entails low negative impact). In this context, the most important criterion for evaluating quality will be the sensitivity—what proportion of positive cases are correctly diagnosed as positive.

It is not feasible to expect the system to anticipate the form of evaluation that will be most appropriate for a particular context. Such a judgment relies on metaknowledge about the domain to which it is not realistic to



Untitled Results

Decision	Decision					Predictive
Selected	primary_l	compensati	secondary_	negative	Total	% Value
primary_hypoth	47	0	0	1	48	97.9
compensated_hy	2	83	0	0	85	97.6
secondary_hypot	0	0	0	0	0	*
negative	0	3	2	1736	1741	99.7
Total Selected	49	86	2	1737	1874	99.6
% Correct	95.9	96.5	0.0	99.9	99.6	
No Decision	6	21	0	8	35	
% Cover	89.1	80.4	100.0	99.5	98.2	
% Sensitivity	85.5	77.6	0.0	99.5	97.7	
% Specificity	99.9	99.9	100.0	97.0		

Fig. 2. Rule set performance summary display.

expect a computer-based system to have access. Such a judgment is best left to the domain expert using the system.

In the light of these factors, it is important to provide more comprehensive evaluation of rule set performance than is traditional in the field of machine learning. It is relatively straight forward to construct a tabular representation of rule set performance that encompasses both predictive accuracy of the rule set as a whole and such measures as positive predictive value, sensitivity and specificity of the rule set with respect to individual classes of outcome. Such a mechanism serves to provide a useful overview of progress for the expert during system development. Fig. 2 displays an example of evaluation provided by The Knowledge Factory. Analysis radiates out from a central presentation of the raw data: a two-dimensional matrix indexed by the cases' correct classifications and the system's classifications containing the raw number of cases in each cell. To the right of this central matrix, the total number of cases assigned by the system to each class is presented followed by the predictive value that this represents (the number of cases correctly assigned to the class as a percentage of the number of cases assigned to the class). Below the raw data are the

- total number of cases belonging to the class for which a classification has been assigned;
- percentage of these for which the correct classification was assigned;
- number of cases belonging to the class for which no classification was made;
- percentage of cases belonging to the class for which a classification was made;
- sensitivity (the percentage of cases belonging to the class for which the correct classification was assigned);
- specificity (the percentage of cases that do not belong to the class that were not classified as belonging to that class).

6.4. Interactive rule set walk through

Example cases can also be used to demonstrate the interactions of rules by stepping through the application of a rule set to an example case. Such a facility greatly enhances the ability of the human expert to gain a detailed understanding of the knowledge base. It also serves to demonstrate how the rules within a rule set interact. The Knowledge Factory allows the user to enter such an interactive session at any stage.

6.5. Case-based expert initiated communication

Cases not only serve as a vehicle for the machine learning system to communicate with the human expert, but they also provide a natural and powerful means for the human to communicate with the induction system.

Where an expert is unable to specify exactly how a rule should be changed, but is aware of deficiencies, the provision of examples provides a simple mechanism for expressing those deficiencies. Counterexamples can be used to demonstrate errors in a rule while positive examples provide a means for expressing how a rule should be extended. Machine learning systems employ such information as normal. Experts are typically accustomed to the use of examples as an aid to the communication of concepts and, in informal studies, have little difficulty in adopting this form of communication.

Any such examples developed by the expert need only be added to the training set for the machine learning system to have complete access to their full import.

As the description of detailed examples can be tedious, it is advantageous if the expert is able to provide partial examples, with irrelevant details left unspecified. This implies, of course, that the machine learning system must be able to handle such partial examples.

It would be possible to further extend the power of case-based communication by allowing the use of invalid examples. These are examples of cases that cannot occur. For instance, if an expert believed that a particular combination of symptoms could not be associated with disease X, but did not wish to provide an example with a particular outcome that was incompatible with X (assuming that 'not X' was not a possible conclusion in the knowledge base under construction), it would be possible to provide an invalid example with the relevant symptom and the diagnosis X, thereby communicating the desired information to the system.

Such a mechanism coupled with the use of partial examples could provide a powerful mechanism for expressing constraints. For example, to communicate that males cannot be pregnant, it is only necessary to create an invalid example which is male and pregnant and for which all other values are unspecified.

In effect, normal examples specify portions of the space of possible knowledge bases that are desirable whereas invalid examples would specify sections of the space of possible knowledge bases that are undesirable. Covered, uncovered, positive and negative examples provide expressions of how a particular knowledge base relates to the specification provided by the set of examples.

Although The Knowledge Factory supports the interactive specification of examples and partial examples, it does not currently support invalid examples.

While case-based communication is powerful, flexible and easy to use, it is difficult to use it for expressing complex constraints and background knowledge. For example, there is no simple manner in which example cases can be used to express a complex relationship between variables such as $E = mc^2$. At best, they will be able to describe points along such a function. If it is not possible to express such interrelationships between variables in the target language, and it is relevant, it would be necessary to use a formal metalanguage in order to communicate it.

7. Other communication mechanisms

While case-based communication is extremely powerful, it is not able to support all aspects of communication required in collaborative knowledge acquisition. A number of additional communications mechanisms have been developed and evaluated within The Knowledge Factory.

7.1. Rule annotations

One simple mechanism for communication about the target language that has been developed for use in The Knowledge Factory is the addition of rule annotations. All communication in The Knowledge Factory is centered around a draft knowledge base. This knowledge base consists of a set of rules expressed in the target production rule language. Each of these rules is annotated with a simple qualitative evaluation: *good*, indicating that the rule is considered of high quality and should not be modified, *revisable*, indicating that the rule is a hypothesis of intermediate quality that is not considered immutable, or *bad*, indicating that the rule is considered unacceptable.

As described above, these annotations are used by the machine learning system during induction. Good rules are considered sacrosanct and are never modified by the machine learning system. Revisable rules may be freely modified during induction. Bad rules are deleted during induction.

In addition to the simple evaluation, further annotation describing the reasons for a particular evaluation can be useful. The current implementation provides only explanations of why the machine learning system has changed an evaluation. The machine learning system has no ability to reason about the human partner's motives for specifying a particular evaluation. Thus, any evaluation provided by the user is described as user evaluation.

If the machine learning system rejects a rule due to it misclassifying example cases, the annotation *invalid* is provided. If a rule is rejected because its inclusion in the rule set does not affect the expert system's performance, the annotation *redundant* is provided.

The machine learning component is never able to set a rule's evaluation to *good*, and so no annotations explaining *good* evaluations are supported.

As it is difficult to provide a meaningful yet succinct explanation as to why the machine learning component should propose any particular new rule, no explanations are provided for revisable evaluations (the evaluation provided by the machine learning component for rules that it induces) either.

This extremely simple rule annotation mechanism is easy to master and it allows simple communication about potential rules for inclusion in the knowledge base. However, like case-based communication, it does not allow communication of complex background knowledge or constraints.

7.2. Communication about collaboration

So far, we have examined mechanisms that allow the partners to communicate about the knowledge base under development. However, collaboration requires communication not only about the objective but also about the means by which that objective is to be reached. Thus, there is a need to communicate about planning or, at very least, control of the collaborative work. Again, a formal language capable of supporting complex task planning and control would be extremely complicated and its use would create a major barrier to machine–expert collaboration for knowledge acquisition.

The Knowledge Factory places the initiative firmly in the hands of the human partner. All long term planning is left entirely to the human partner. All control communication takes the form of commands issued by the human partner to the machine learning system. For

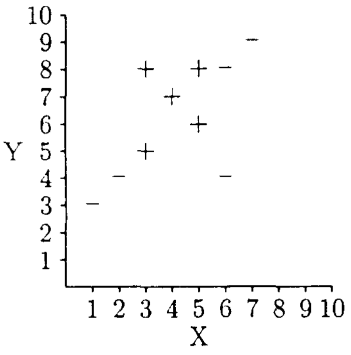


Fig. 3. Simple attribute space with two attributes.

ease of use, these commands are issued via a standard menu and dialogue interface.

However, there is clearly great potential value in allowing the machine learning component to assume the initiative in opportune circumstances. For example, if the system were to observe the human partner making a series of changes to the knowledge base that led to a decrease in performance, it might be opportune for it to notify the human partner of this and to suggest steps that might rectify the situation.

In the belief that it is important for the human partner to feel in control of the joint project, such computer generated seizures of initiative should not be obtrusive and should take the form of suggestions rather than commands. Thus, the computer should wait until the human partner is not engaged in an important operation before taking the initiative and all such initiatives should be subject to approval by the human partner before any irrevocable action occurs. Further, such actions should be kept to a minimum and should only occur when the potential gains are substantial.

As the forms of computer generated initiative are likely to be restricted, a simple dialogue mechanism should suffice for communication in this context. While such mechanisms are not currently implemented in The Knowledge Factory, their investigation is regarded as a promising direction for future research.

8. Alternative rules

One way to envisage the operation of a machine learning system is as the partitioning of an attribute space. Each partition is labeled with a class. For The Knowledge Factory's machine learning system, each partition corresponds to a classification rule. One of the most important aspects of a partition will be the cluster of cases of its class that it separates from other cases. For some machine learning problems, the clusters of cases that should form a given partition will be straightforward to distinguish, but the precise partition with which that cluster should be partitioned will not. Consider a simple attribute space as depicted in Fig. 3. This attribute space is defined by two attributes, X and Y. Five positive cases are each indicated by + symbols and five negative cases are each indicated by − symbols. It seems apparent that there is a single cluster of positive examples. What is far from apparent, however, is exactly how these examples should be partitioned from the negative cases. Any of the partitions

$X \leqslant 5 \wedge Y \geqslant 5$,

$3 \leqslant X \leqslant 5$

or

$3 \leqslant X \leqslant 5 \wedge Y \geqslant 5$

could equally be employed. Without access to additional information of a type not readily available to a machine learning system, there is no way to select between these partitions. The machine learning system's selection is necessarily arbitrary.

This would appear to be a good issue on which to receive advice from the expert. It is quite possible that the expert's knowledge will enable him or her to make an informed selection between the alternatives on the basis of knowledge external to the formal considerations available to the machine learning system.

However, it may also be the case that the expert has no means for selecting between these partitions. Rather than requiring the expert to always select from the available partitions, The Knowledge Factory provides the user with the option of requesting alternatives to an existing partition from which he or she may select an alternative.

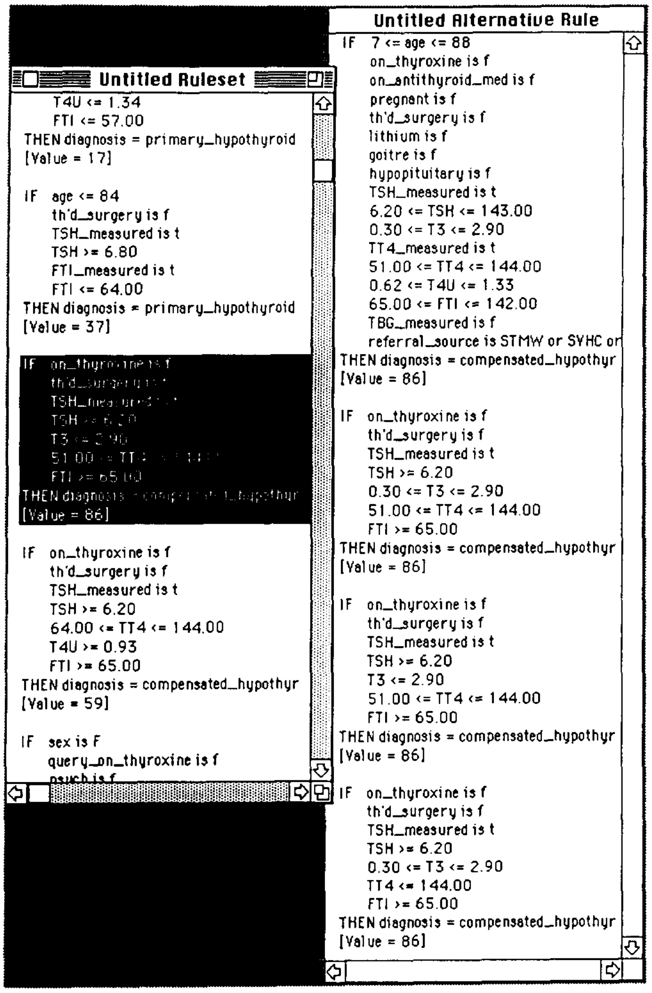
The Knowledge Factory generates a set of alternative partitions for a rule r as follows.

First it generates the least generalisation of all the positive examples covered by r. This is the most specific rule s that covers all of those cases covered by r.

Next it generates all the greatest conjunct deletion generalisations of s that do not cover a negative case not covered by r. A conjunct deletion generalisation of a rule x is a rule formed by deleting one or more conjuncts from x. A rule g is a greatest conjunct deletion generalisation of a rule x that does not cover a case c if it is a conjunct deletion generalisation of x that does not cover c and there is no conjunct deletion generalisation of g that does not cover c. The greatest conjunct deletion generalisations are highly general alternatives to r.

Finally, it generates the conjunction of all the greatest conjunct deletion generalisations. This is an alternative of intermediate generality.

Fig. 4 shows a set of alternative rules. The highlighted rule is the original rule for which alternatives have been generated. The first rule in the alternative rules window is the least generalisation of all positive examples covered by the original rule. The second rule is the conjunction of all the greatest conjunct deletion generalisations. The remaining rules are the greatest conjunct deletion



```
    T4U <= 1.34
    FTI <= 57.00
THEN diagnosis = primary_hypothyroid
[Value = 17]

IF  age <= 84
    th'd_surgery is f
    TSH_measured is t
    TSH >= 6.80
    FTI_measured is t
    FTI <= 64.00
THEN diagnosis = primary_hypothyroid
[Value = 37]

IF  on_thyroxine is f
    th'd_surgery is f
    TSH_measured is t
    TSH >= 6.20
    T3 <= 2.90
    51.00 <= TT4 <= 144.00
    FTI >= 65.00
THEN diagnosis = compensated_hypothyr
[Value = 86]

IF  on_thyroxine is f
    th'd_surgery is f
    TSH_measured is t
    TSH >= 6.20
    64.00 <= TT4 <= 144.00
    T4U >= 0.93
    FTI >= 65.00
THEN diagnosis = compensated_hypothyr
[Value = 59]

IF  sex is F
    query_on_thyroxine is f
    psych is f
```

```
IF  7 <= age <= 88
    on_thyroxine is f
    on_antithyroid_med is f
    pregnant is f
    th'd_surgery is f
    lithium is f
    goitre is f
    hypopituitary is f
    TSH_measured is t
    6.20 <= TSH <= 143.00
    0.30 <= T3 <= 2.90
    TT4_measured is t
    51.00 <= TT4 <= 144.00
    0.62 <= T4U <= 1.33
    65.00 <= FTI <= 142.00
    TBG_measured is f
    referral_source is STMW or SVHC or
THEN diagnosis = compensated_hypothyr
[Value = 86]

IF  on_thyroxine is f
    th'd_surgery is f
    TSH_measured is t
    TSH >= 6.20
    0.30 <= T3 <= 2.90
    51.00 <= TT4 <= 144.00
    FTI >= 65.00
THEN diagnosis = compensated_hypothyr
[Value = 86]

IF  on_thyroxine is f
    th'd_surgery is f
    TSH_measured is t
    TSH >= 6.20
    T3 <= 2.90
    51.00 <= TT4 <= 144.00
    FTI >= 65.00
THEN diagnosis = compensated_hypothyr
[Value = 86]

IF  on_thyroxine is f
    th'd_surgery is f
    TSH_measured is t
    TSH >= 6.20
    0.30 <= T3 <= 2.90
    TT4 <= 144.00
    FTI >= 65.00
THEN diagnosis = compensated_hypothyr
[Value = 86]
```

Fig. 4. Set of alternative rules.

generalisations of the original rule. It should be noted that this example was chosen for ease of presentation because it contained only a small number of alternative rules. In our experience, for most rules, there are more than the two greatest conjunct deletion generalisations shown in this example. It is interesting to observe that machine learning systems appear to regularly be placed in the position of arbitrarily selecting between alternative rules with equal support.

The user can replace the original version of the rule with any selection from this list. Alternatively, he or she can add one or more of the alternatives to the current rule set.

Any rule that is a generalisation of the least generalisation of all the positive examples of a rule r and a specialisation of a greatest conjunct deletion generalisation of r will cover the same positive cases as r and no negative cases not covered by r. It is not feasible to generate and display all of these alternatives due to the potential combinatorial explosion in their number (in the example presented in Fig. 4 there are 32 768 such alternatives). The user may wish to select a rule from this space of alternatives by selecting one of the greatest conjunct deletion generalisations and then adding clauses from the least general rule.

9. Automated rule modification

In some circumstances a user will want a particular rule to perform in a particular manner but will not know how best to produce that performance. For example, there might be a number of cases that are not covered by a rule that the user feels should be covered by it, but the user is not sure how to achieve this end. The Knowledge Factory allows the user to experiment with modifications to a rule that are specified by desired outcome rather than by explicit changes to the rule. Two types of desired outcome are supported: generalising the rule to cover identified cases, and specialising the rule to not cover specified cases.

When generalising to cover a case, the least generalisation [40] of the rule against the case is formed. This ensures that the resulting rule covers no more negative cases than necessary.

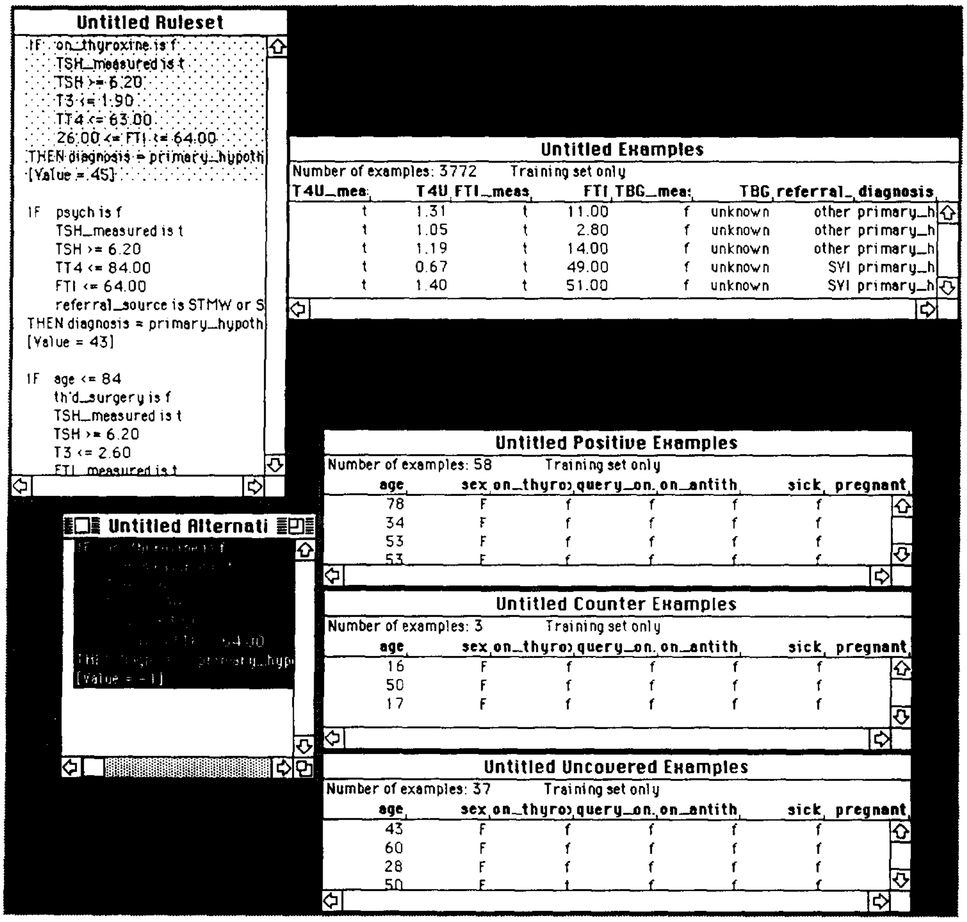
When specialising to exclude a case, all least specialisations of the rule against the case are formed. As there will normally be many such rules, all are generated and the user is able to evaluate and select between them. As with the least generalisation, the least specialisation is a modification of least commitment, modifying the original rule the least amount necessary to exclude the nominated case.

Once the generalisations or specialisations are formed, the user can explore further variations thereof, by direct editing, generalising or specialising against other specific cases, or generating alternative rules. If and when the user is satisfied with the outcome, the original rule can be replaced by the alternative or the alternative can be added to the rule set.

Fig. 5 shows a rule (the first rule displayed in the *untitled ruleset* window), a case against which it has been generalised (the first case in the *untitled examples* window), the resulting rule (in the *untitled alternative rules* window), and examples windows evaluating the resulting rule.

10. Experience using the system

Except where explicitly stated otherwise, The Knowledge Factory incorporates all of the facilities described above. This system has been used by a number of distinct groups of users. The two contexts in which major projects have been conducted are financial analysis and medical diagnosis. The users in the first case are experienced computing professionals while the users in the second case have only basic exposure to computers from an end user perspective. In both contexts the system has



Untitled Ruleset

```
IF   on_thyroxine is f
     TSH_measured is t
     TSH >= 6.20
     T3 <= 1.90
     TT4 <= 63.00
     26.00 <= FTI <= 64.00
THEN diagnosis = primary_hypoth
[Value = 45]

IF   psych is f
     TSH_measured is t
     TSH >= 6.20
     TT4 <= 84.00
     FTI <= 64.00
     referral_source is STMW or S
THEN diagnosis = primary_hypoth
[Value = 43]

IF   age <= 84
     th'd_surgery is f
     TSH_measured is t
     TSH >= 6.20
     T3 <= 2.60
     FTI_measured is t
```

Untitled Examples

Number of examples: 3772 Training set only

T4U_mea	T4U	FTI_meas	FTI	TBG_mea	TBG	referral_	diagnosis
t	1.31	t	11.00	f	unknown	other	primary_h
t	1.05	t	2.80	f	unknown	other	primary_h
t	1.19	t	14.00	f	unknown	other	primary_h
t	0.67	t	49.00	f	unknown	SVI	primary_h
t	1.40	t	51.00	f	unknown	SVI	primary_h

Untitled Positive Examples

Number of examples: 58 Training set only

age	sex	on_thyro	query_on.	on_antith	sick	pregnant
78	F	f	f	f	f	
34	F	f	f	f	f	
53	F	f	f	f	f	
53	F	f	f	f	f	

Untitled Counter Examples

Number of examples: 3 Training set only

age	sex	on_thyro	query_on.	on_antith	sick	pregnant
16	F	f	f	f	f	
50	F	f	f	f	f	
17	F	f	f	f	f	

Untitled Uncovered Examples

Number of examples: 37 Training set only

age	sex	on_thyro	query_on.	on_antith	sick	pregnant
43	F	f	f	f	f	
60	F	f	f	f	f	
28	F	f	f	f	f	
50	F	t	f	f	f	

Fig. 5. A rule and its expansion.

been readily mastered by the users with successful expert systems being constructed. Both groups of users have expressed satisfaction with the facilities provided by the system.

The system has also been used extensively for smaller knowledge acquisition projects, primarily by third year university computer science students. A study was conducted using 16 students in a third year artificial intelligence and expert systems unit. After six weeks studying expert systems, including an introduction to the principles of expert systems and programming in CLIPS, these students were each given as an assignment the task of using The Knowledge Factory to create an expert system for an artificial medical diagnosis domain. Each student was given, in written form, a body of background knowledge. This was augmented by a set of 250 example cases. From this, the students had to each create an expert system. The students were each given a manual and a half hour introduction to the use of the system was given in class. At the end of this two week project the students were given a questionnaire examining their experience. This questionnaire included seven questions rated on a scale from 1, representing *not at all*, to 5, representing *very*. The average responses to these questions are listed in Table 1.

It is clear that these students found the system easy to use, despite the limited training that they were given in its use. They regarded the system to be a valuable tool and believed that machine learning was a valuable tool for knowledge acquisition. The rule editing facilities were regarded as easy to use, although it should be borne in mind that third year computer science students should be proficient at editing in general. The students found the task slightly difficult. It is not clear how to interpret this

Table 1
Questionnaire results

Question	Rating
How easy was The Knowledge Factory to use?	4.4
How difficult was it to create an expert system for this assignment?	2.3
How difficult was it to edit rules?	1.6
How easy was it to assess the quality of individual rules using The Knowledge Factory?	3.6
How easy was it to assess the quality of a set of rules using The Knowledge Factory?	3.7
How valuable do you think machine learning is to knowledge acquisition?	4.5
Do you think that The Knowledge Factory is a valuable tool for building expert systems?	4.4

result, however, as it is not possible to distinguish difficulty introduced by the system from general task difficulty. There appears to have been some ambivalence about the ease with which it was possible to assess the quality of both individual rules and sets of rules. Again, it is not possible to distinguish here between the ease of difficulty of the use of the facilities provided and the ease or difficulty of rule quality assessment in general.

In all, these results are positive in that they make clear that users without extensive training as knowledge engineers can readily master the tool. These results can be considered as a proof of concept for the proposition that non-knowledge-engineers can readily collaborate with a machine learning system to develop expert systems. There is still room, however, to perform comparative evaluations of the relative merits of alternative approaches to this form of knowledge acquisition.

11. Conclusions

Three decades of intensive research into machine learning has seen impressive results and the development of numerous useful automated knowledge acquisition tools. However, an autonomous machine learning system will always be limited by the comprehensiveness of its training set. Unless every relevant combination of factors is represented in the training set, no matter how infrequently it occurs in practice, a machine learning system cannot be expected to produce a perfectly accurate knowledge base. As it will frequently be the case that a training set will not be sufficiently comprehensive, autonomous machine learning is necessarily limited in the extent of its applicability.

Nevertheless, even though a machine learning system will not be able to produce a perfect knowledge base from an incomplete training set, it may still be able to derive valuable insights therefrom. Frequently, these insights will be quite different from those otherwise available to the human expert.

In consequence, there is every reason to believe that machine learning is able to provide an adjunct source of insight during the knowledge-acquisition process. Machine–expert collaboration for knowledge acquisition provides one approach to harnessing that insight.

Techniques for machine–expert collaboration for knowledge acquisition are still in their infancy. This paper has identified three key issues: control, capabilities and communication. It is argued that control should be placed in the hands of the human expert, who is likely to be situated in the context in which the knowledge base is to be employed and who will have considerable influence, once knowledge acquisition is completed, upon the success or failure of its application. However, although the expert should have control, both parties should be able to provide initiative.

The range of capabilities that are required is likely to depend greatly on the context. However, it is essential that the machine learning system be able to refine successive drafts of the knowledge base and that the human expert be able to perform arbitrary changes to the knowledge base and provide advice and guidance to the machine learning system.

The communication facilities should be easy to master while allowing the expression of complex knowledge, metaknowledge and cooperative control information. It is appropriate to use a number of distinct languages for communication. Knowledge can be expressed in the target language. Simple metaknowledge can be expressed by annotations to expressions in the target language. Complex knowledge and metaknowledge can be expressed through the use of example cases. Finally, a simple command language can be used to manage the process of collaboration.

These facilities are well within the reach of current technology. Indeed, most are implemented in the The Knowledge Factory system. The Knowledge Factory allows the user to

- edit any visible rule or example in situ,
- explore the quality of individual rules, and, in particular, explore their relationship to cases in the training set,
- evaluate the performance of the rule set on the training set,
- explore alternatives to individual rules,
- provide qualitative evaluation of individual rules,
- add new examples,
- revise the metamodel,
- apply the knowledge base in an interactive environment,
- provide direction to the machine learning component.

All of these mechanisms are readily assimilated, even by users with relatively little computer experience and no knowledge acquisition experience. Although these communications mechanisms are simple, as outlined above, they support extremely powerful dialogue.

Most importantly, the key mechanisms are sufficiently familiar for untrained users to employ them with little or no tuition. New users with no previous exposure to knowledge acquisition rapidly enter into dialogue with The Knowledge Factory without even having apparent conscious awareness of the deep messages being carried by the simple surface interactions. For instance, it is so natural for a human expert to provide a counterexample that he or she does not need to explicitly consider the deep message that it conveys (that the rule under critique is deficient in that it cannot accommodate this new case and that The Knowledge Factory should do something about this deficiency).

While much remains to be done, the integration of machine learning with knowledge elicitation in a form that can be readily employed directly by a domain expert has been demonstrated to be both practical and effective.

Acknowledgments

This research has been supported by the Australian Research Council and the Apple University Development Fund.

I am grateful to Jason Wells for valuable comments on earlier drafts of this paper and for assistance in the development of The Knowledge Factory software. I am also grateful to Tim Menzies for encouraging me to use undergraduate students for system evaluation.

References

[1] D.A. Waterman, A Guide to Expert Systems, Addison–Wesley, USA, 1986.

[2] J.W.M. Agar and G.I. Webb, The application of machine learning to a renal biopsy data-base, Nephrology, Dialysis & Transplantation, 7 (1992) 472–447.

[3] Structured Decision Tasks Methodology for Developing and Integrating Knowledge Base Systems, Attar Software, Leigh, Lancs., UK, 1989.

[4] R. Davis and D.B. Lenat, Knowledge-Based Systems in Artificial Intelligence, McGraw–Hill, USA, 1982.

[5] L. De Raedt, Interactive Theory Revision, Academic Press, UK, 1992.

[6] K. Morik, S. Wrobel, J. Kietz and W. Emde, Knowledge Acquisition and Machine Learning: Theory, Methods and Applications, Academic Press, UK, 1993.

[7] C. Nedellec and K. Causse, Knowledge refinement using knowledge acquisition and machine learning methods, in Proc. EKAW '92: Current Developments in Knowledge Acquisition, Springer-Verlag, Germany, 1992, 171–190.

[8] J.L. O'Neil and R.A. Pearson, A development environment for inductive learning systems, in Proc. 1987 Australian Joint Artificial Intelligence Conf., Sydney, Australia, 1987, pp. 673–680.

[9] R.G. Smith, H.A. Winston, T.M. Mitchell and B.G. Buchanan, Representation and use of explicit justifications for knowledge base refinement, in Proc. Ninth Int. Joint Conf. Artificial Intelligence, Morgan Kaufmann, USA, 1985, pp. 673–680.

[10] G. Tecuci and Y. Kodratoff, Apprenticeship learning in imperfect domain theories, in Y. Kodratoff and R.S. Michalski (eds.), Machine Learning: An Artificial Intelligence Approach, Morgan Kaufmann, USA, 1990, pp. 514–551.

[11] D.C. Wilkins, Knowledge base refinement using apprenticeship learning techniques, in Proc. AAAI-88: Seventh National Conf. Artificial Intelligence, Morgan Kaufmann, USA, 1988.

[12] P. Compton, G. Edwards, A. Srinivasan, R. Malor, P. Preston, B. Kang and L. Lazarus, Ripple down rules: turning knowledge acquisition into knowledge maintenance, Artificial Intelligence in Medicine, 4 (1992) 47–59.

[13] J.H. Boose, ETS: a system for the transfer of human expertise, in J.S. Kowalik (ed.) Knowledge Based Problem Solving, Prentice–Hall, USA, 1986.

[14] L. Eshelman, D. Ehret, J. McDermott and M. Tan, MOLE: a tenacious knowledge acquisition tool, Int. J. Man–Machine Studies, 26 (1987) 41–54.

[15] G. Kahn, MORE: from observing knowledge engineers to automating knowledge acquisition, in S. Marcus (ed.), Automating Knowledge Acquisition for Expert Systems, Kluwer, USA, 1988, pp. 7–35.

[16] S. Marcus, SALT: a knowledge-acquisition tool for propose-and-revise systems, in S. Marcus (ed.), Automating Knowledge Acquisition for Expert Systems, Kluwer, USA, 1988, pp. 81–123.

[17] E. Bloedorn and R.S. Michalski, Data-driven constructive induction in AQ17-PRE: a method and experiments, in Proc. 1991 IEEE Int. Conf. Tools for Artificial Intelligence, IEEE, USA, 1991, pp. 30–37.

[18] G. Drastal, S. Raatz and G. Czako, Induction in an abstract space: a form of constructive induction, in Proc. IJCAI-89, Morgan Kaufmann, USA, 1989, pp. 708–712.

[19] R. Elio and L. Watanabe, An incremental deductive strategy for controlling constructive induction in learning from examples, Machine Learning, 7 (1991) 7–44.

[20] L. Fu and B.G. Buchanan, Learning intermediate concepts in constraining a hierarchical knowledge base, in Proc. IJCAI-85, Los Angeles, CA, USA, 1985, pp. 659–666.

[21] C.J. Matheus and L.A. Rendell, Constructive induction on decision trees, in Proc. IJCAI-89, Morgan Kaufmann, USA, 1989, pp. 645–650.

[22] P. Mehra, L.A. Rendell and B.W. Wah, Principled constructive induction, in Proc. IJCAI-89, Morgan Kaufmann, USA, 1989, pp. 651–656.

[23] S. Muggleton, Duce, an oracle based approach to constructive induction, in Proc. IJCAI-87, Morgan Kaufmann, USA, 1987, pp. 287–292.

[24] S.K. Murthy, S. Kasif and S. Salzberg, A system for induction of oblique decision trees, J. Artificial Intelligence Research, 2 (1994) 1–32.

[25] G. Pagallo and D. Haussler, Boolean feature discovery in empirical learning, Machine Learning, 5 (1990) 71–100.

[26] L. Rendell and R. Seshu, Learning hard concepts through constructive induction: framework and rationale, Computational Intelligence, 6 (1990) 247–270.

[27] S.M. Weiss and N. Indurkha, Reduced complexity in rule induction, in Proc. IJCAI-91, Morgan Kaufmann, USA, pp. 781–787.

[28] J. Wnek and R.S. Michalski, Hypothesis-driven constructive induction in AQ17-HCI: a method and experiments, Machine Learning, 14 (1994) 139–168.

[29] J. Wogulis and P. Langley, Improving efficiency by learning intermediate concepts, in Proc. IJCAI-89, Morgan Kaufmann, USA, 1989, pp. 657–662.

[30] S. Yip and G.I. Webb, Discriminant attribute finding in classification learning, in A. Adams and L. Sterling (eds.), Proc. AI '92, Singapore, 1972, pp. 374–379.

[31] Z.-J. Zheng, Constructing conjunctive tests for decision trees, in A. Adams and L. Sterling (eds.), Proc. AI '92, Singapore, 1992, pp. 355–360.

[32] W.J. Clancey, Model construction operators, Artificial Intelligence, 53 (1992) 1–115.

[33] D. Ourston and R.J. Mooney, Changing the rules: a comprehensive approach to theory refinement, in Proc. AAAI-90, pp. 815–820.

[34] G.I. Webb, DLGref2: techniques for inductive knowledge refinement, in Proc. IJCAI Workshop W16, Chambery, France, 1993, pp. 236–252.

[35] G.I. Webb and J.W.M. Agar, Inducing diagnostic rules for glomerular disease with the DLG machine learning algorithm, Artificial Intelligence in Medicine, 4 (1992) 314.

[36] K. Morik, Sloppy modeling, in K. Morik (ed.), Knowledge Representation and Organization in Machine Learning, Springer-Verlag, USA, 1989, pp. 107–134.

[37] L.G. Terveen, Overview of human–computer collaboration, Knowledge-Based Systems, 8 (1995) 67–81.

[38] C. Sammut and R.B. Banerji, Learning concepts by asking questions, in R.S. Michalski, J.G. Carbonell and T.M. Mitchell, (eds.), Machine Learning: An Artificial Intelligence Approach, Vol. II, Morgan Kaufmann, USA, 1986, pp. 167–191.

[39] S.M. Weiss, R.S. Galen and P. Tadepalli, Maximizing the predictive value of production rules, Artificial Intelligence, 45 (1990) 47–71.

[40] G.D. Plotkin, A note on inductive generalisation, in B. Meltzer and D. Mitchie (eds.), Machine Intelligence 5, Edinburgh University Press, UK, 1970, pp. 153–163.