Pre-publication draft of a paper which appeared in the Proceedings of the Eighth Australian Joint Conference on Artificial Intelligence (AI'95) pp 291-298

Recent progress in machine-expert collaboration for knowledge acquisition.

Geoffrey I Webb
Jason Wells

School of Computing and Mathematics
Deakin University
Geelong 3217 Australia
{webb|wells}@deakin.edu.au

Abstract

Knowledge acquisition remains one of the primary constraints on the development of expert systems. A number of researchers have explored methods for allowing a machine learning system to assist a knowledge engineer in knowledge acquisition. In contrast, we are exploring methods for enabling an expert to directly interact with a machine learning system to collaborate during knowledge acquisition. We report recent extensions to our methodology encompassing a revised model of the role of machine learning in knowledge acquisition; techniques for communication between a machine learning system and a domain expert and novel forms of assistance that a machine learning system may provide to an expert.

Keywords: Machine Learning; Knowledge Acquisition; Knowledge Elicitation

Introduction

Despite two decades of research, knowledge acquisition remains a primary bottleneck for expert system development. The two primary approaches to knowledge acquisition are knowledge elicitation from experts and machine learning. Each has distinct strengths and weaknesses.

Knowledge elicitation involves the encoding of knowledge that is expressed by a human expert. This knowledge takes the form of a model of the domain that is constructed by the expert, often with assistance from a knowledge engineer or automated assistant. Human experts are able to draw upon book learning (the accumulated wisdom of thousands of years of human endeavour), practical experience, general and common-sense knowledge and are situated in the social and operational context in which the knowledge is to be employed.

Machine learning involves formal automated analysis of example cases from which a knowledge-base (model of the domain) is inferred. 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 human biases lead experts to overlook; and
- may induce knowledge in contexts in which human expertise is unable to provide insight, for example, because no one has yet developed solutions to the particular problem.

Conversely, human expertise—

- may be able to provide solutions in circumstances for which the available example cases are not adequate for a machine learning system to derive a suitable solution; and
- will often be able to discriminate between potential solutions on the basis of knowledge that is external to the formal considerations available to the machine learning system.

Further, a machine learning system is constrained by the problem description that is provided. If a particular factor is necessary for correct analysis of a domain and the machine learning system is not provided with knowledge of that factor then it is not possible for the machine learning system to produce a correct analysis. The *domain model* (set of primitive terms, predicates and operators with which the system is provided) defines the space of possible solutions that it can generate. If an adequate solution does not lie within that space then it is not possible for the machine learning system to find one. Although there has been considerable research into techniques for extending domain models by learning new terms and predicates [1, 2, 3, 4] the new terms and predicates so learned are necessarily derived from the primitive terms and predicates with which the system is provided. Such approaches extend the space of possible solutions but still fail to enable a machine learning system to develop adequate solutions when the initial domain model lacks key terms, predicates or operators. Morik [5] has investigated techniques for machine learning systems to identify deficient domain models but still requires knowledge elicitation to rectify the identified deficiencies.

In view of the differing and complementary capabilities of knowledge elicitation and machine learning, there is considerable potential for gain through the integration of the two approaches. A number of research groups have investigated the integration of machine learning and knowledge elicitation [6, 7, 8, 9]. These approaches provide environments suited to sophisticated knowledge engineers. In contrast, the current research investigates the integration of knowledge elicitation with machine learning for users with little or no expertise in knowledge acquisition or knowledge based systems [10]. In such a context it is important that the knowledge representation language and the interactions between the system and the user be simple to use and master.

Webb [11] described initial work toward the development of techniques to support such interaction. This paper describes our recent progress in refining and extending these techniques.

Overview

Our attention is focused on a very specific type of knowledge acquisition task—those:

- that can be handled by a domain expert without aid from an expert knowledge engineer;
- for which a domain expert has non-trivial insight; and
- for which it is possible to accumulate a suitable set of example cases.

It can be seen that this type of situation lies between standard machine learning, previous approaches to integrating machine learning with knowledge elicitation and automated knowledge elicitation aids such as ripple-down-rules [12] and repertory grids [13]. It is distinguished from standard machine learning by the tight integration of the domain expert into the knowledge acquisition process. It is distinguished from previous approaches to integrating machine learning with knowledge elicitation 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 removing the reliance on the domain expert's ability to correctly solve problems from the domain.

The system was previously called 'Einstein'. To enable us to distinguish two distinct components of the system—the interactive collaborative interface and the induction engine—the name 'Einstein' is now reserved for the latter and the name 'The Knowledge Factory' is now applied to the former.

The knowledge representation formalism is restricted to simple tests on attribute-values. We contend that sophisticated use of first-order logic would provide a considerable barrier to use by naive users. In particular, we believe that a simple attribute-value representation formalism is appropriate for a system that is to be used without the assistance of a knowledge engineer.

The Knowledge Factory is oriented toward knowledge acquisition tasks for which both a domain expert and a body of examples such as are typically provided to a machine learning system are available. It is expected that both the machine learning component and the domain expert will have differing and complementary insights into the domain. They can apply those insights either by directly modifying the existing knowledge base, or by providing comments to the other.

If the two are to communicate with each other, as the provision of comments implies, there is a need for a suitable language. There are a number of constraints upon such a language. It must be comprehensible to both parties. It must be sufficiently expressive to enable the communication of sophisticated critiques of a knowledge base. In the context of a system aimed at unsophisticated users, it must be simple to master and use.

Our solution is to use examples as the primary means of communication. Examples have meaning to a machine learning system. They provide the reasons for the system's actions (it will develop a particular classifier precisely because it handles the available examples in a particular manner). Examples also capture a machine learning system's objections to a proposed rule. Why would it object to a rule? Because the rule will misclassify particular examples. Further, experts are typically adept at handling example cases. Expertise is, after all, the ability to handle cases appropriately.

The techniques presented by Webb [10] were centred around communication via examples (although this abstract description of the form of communication was not made explicit).

Two primary techniques were developed to allow the machine learning system to critique the existing rule set. The first was through case-based critique—the display of the example cases that a rule correctly classified (positive support for the rule); misclassified (evidence against) and failed to classify (cases that the rule could profitably be generalised to handle). The second technique was the development of revisions to the existing rule set through analysis of the example cases.

Three primary techniques were developed to allow the user to critique the current rule set. One technique was for the user to make qualitative judgements about individual rules—*unacceptable*; *acceptable* (but could be improved); or *accepted* (do not alter). Another technique was for the user to provide critique in the form of examples ("this rule is no good because it misclassifies the following type of case"). Such examples could be incomplete or prototypal. The third technique was through direct manipulation of the rule set ("this rule should be ....").

A number of facilities were provided for the user to manage the induction process. The user could nominate attributes that should be considered in rules for a given class. The user could control the level of generality of the rules inferred. Highly general rules will classify more cases with lower expected accuracy [11].

An important innovation of the system was the ability to modify the domain model at any stage during the knowledge acquisition process. Attributes could be added or deleted. This was organised in such a way as to minimise the disruption to the existing rule set. This contrasts with traditional machine learning for which revisions to the domain model, necessitate that induction start again from the beginning, losing any work performed to date.

Since 1992 we have been working to extend this methodology. The rule refinement techniques have been substantially reworked, as reported elsewhere [14]. The following sections report on the progress that we have made in extending other aspects of the integrated machine learning and knowledge elicitation methodology that we are developing.

Alternative rules

Recent developments in machine learning, and, in particular, our own recent reappraisal of the field [11, 15] have led us to substantially revise our interpretation of the role that a machine learning system might most profitably play in an interactive knowledge acquisition system. Previously, we took the view that the role of the machine learning system, unlike that of the expert, was to develop final rules (rules that require no further refinement). However, we have come to take a slightly unusual view of the role of a machine learning system in knowledge acquisition. We now view the learning task in the context of partitioning a feature space into areas labelled by classes. The machine learning system is seeking a partitioning that maximises a quality evaluation function with respect to the training set. It is restricted in the set of partitionings that it can consider by the language for expressing partitionings with which it is provided. This process has two aspects—identifying clusters of cases that can (and should) be placed within one of the partitions and forming the partitions around those clusters. Most approaches evaluate the quality of a clustering, not the details of the partitions into which the clusters are placed. That is, they consider the appropriateness of the assignment of cases to partitions labelled with a particular class, but not the decision surfaces that are selected to form the partitions. (A notable exception is MML [16].)

In this context, it is quite likely that there will arise situations in which a machine learning system identifies a useful cluster of cases, but selects a partition to enclose that cluster that the domain expert considers inappropriate.

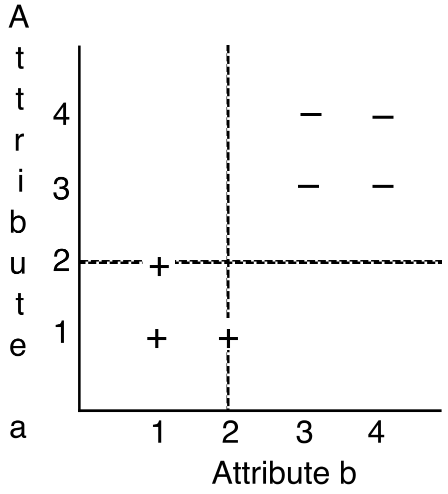

Figure 1: A feature space with multiple credible partitionings for a single cluster of examples

Figure 1 illustrates this situation, with respect to a trivial two attribute domain. The same principles apply, but the situations become much more difficult to evaluate, as the number of attributes increases. In this example, the positive cases (represented by '+') can be partitioned from the negative (represented by '–') by either or both of the simple constraints a ≤ 2 and b ≤ 2. The machine learning system is likely to conclude that all positive cases should be grouped into a single cluster (belong in a single partition). A typical machine learning system will, however, typically make an arbitrary selection of the constraint to employ to form that partition. The selection of constraint may make little difference to the machine learning system, but it is likely to greatly impact on the quality of the classification rule as perceived by the domain expert. Any one of the three constraints (a ≤ 2 or b ≤ 2 or a ≤ 2 & b ≤ 2) may be preferable to the domain expert. There is no principled means for the machine learning system to select which constraint will be preferable, unless it has access to forms of background knowledge not normally employed in machine learning.

Machine learning systems have access to the information needed to provide alternative partitionings for a cluster of cases. There seems to be potential for gain if the user is allowed to make this final decision. By the same token, however, we do not want to burden the user with numerous trivial decisions—in many situations he or she be content to allow the machine learning system to select, for example, because he or she has no greater insight into the appropriate constraint than does the machine learning system. There is a further difficulty in that there will frequently be a large number of constraints, all of which will equally partition a given set of cases from all others.

A restricted set of these constraints can readily and usefully be identified, however. The most specific constraint that covers all cases in a cluster will be of interest because it will include all factors that might be relevant to a partition that isolates those cases. Further, it has been claimed that for appropriate learning tasks, the selection of most specific rules will in general maximise positive predictive value [11].

Most general constraints are also likely to be of interest as these will employ the least numbers of factors and hence be most readily apprehended by the user. There is also evidence that more general classifiers lead to greater predictive accuracy [11].

Finally, a conjunction of all most general constraints will provide a constraint of intermediate generality and complexity that eliminates reference to any factors that are clearly not required in order to partition the cluster of cases from all others.

Taking all these factors into account, the cleanest technique for supporting this type of facility would appear to be to allow the machine learning system to select an individual constraint as normal, but to allow the user to request a list of alternatives and to replace the machine learning system's selection with one of these alternatives if so desired. Figure 2 illustrates the interface that we have provided to support such an interaction. Each rule represents a constraint that can partition the cluster of positive examples covered by the current rule from all negative examples not covered by that rule. A menu command allows the user to request the generation of such a list of alternative rules. The highlighted rule is the original for which alternatives have been requested. The window in the background presents the four alternative rules that have been proposed by the machine learning component. The first is the most specific rule. The second rule is of intermediate generality. The last two are the most general rules.

Figure 2: Presentation of multiple credible partitions for a single cluster of examples

Altering the cluster

While this facility allows the user to readily select between alternative partitions for a given cluster of cases, it will also often be the case that the user wishes to modify the actual clusters that are to form the basis of individual partitions. That is, in the context of the two layer model of machine learning identified above, the domain expert may wish to critique the actions of the machine learning system in either layer. To this end we believe that it is useful to provide facilities to enable the user to require the inclusion or exclusion of specific cases from a cluster.

A simple means to extend a rule to cover a single additional case is least generalisation [17]. We believe that this provides a suitable mechanism for extending a partition to include a case as it extends that partition the least possible amount.

The counterpart to least generalisation, for excluding cases from a partition, is least specialisation. However, while there will always be a single least generalisation of a single rule with regard to a single case within the language for expressing rules with which we work, there may be multiple least specialisations. Our solution is to provide a list of all least specialisations from which the user may select one.

These mechanisms extend the range of case-based interactions that are provided to the user. We believe that these mechanisms provide powerful facilities for a naive user to refine rule sets without the need to articulate specific modifications. As previously argued, the articulation of specific modification is often difficult for the domain expert whereas the use of examples to illustrate required changes may not be so difficult.

These new mechanisms can be coupled with the previously identified technique of critiquing a rule set by specifying counter-examples. If the user feels that a rule is deficient because it fails to accommodate an example case that is not provided in the training set, he or she can readily specify such an example and then use the new mechanisms to require that the rule accommodate that example. Similarly, if a rule will reach an incorrect conclusion, the user can specify a counter-example and require the system to modify the rule to exclude that case. The use of these mechanisms can result in an interactive training process similar to that of MARVIN [18].

Providing general guidance on the forms of partition to select

The previous version of our system allowed the user to select between three different levels of generality in the rules the machine learning system inferred. Reappraised in terms of our more recent model of the role of a machine learning system, these correspond to the most specific, the most general and an intermediate level of rules that covered each cluster of examples.

Given our improved understanding of the implications of varying the generality of inferred rules [11], and of the potential for varying misclassification costs for different classes, we can now see the value of allowing finer grained control over the level of generality of inferred rules. In particular, we now believe that it will often be valuable to be able to request different levels of generality in the rules inferred for different classes.

For example, consider a hypothetical domain with three classes, A, B and C for which there is a high cost for misclassifying cases of A as belonging to other classes and a high cost for misclassifying cases of other classes as belonging to class C. In this situation it would be appropriate to infer highly general rules for class A; rules of intermediate generality for class B; and highly specific rules for class C. Based on the implications of varying levels of generality predicted by Webb [11], this should result in the minimisation of high cost misclassification in this hypothetical situation.

The Knowledge Factory now provides facilities for allowing the user to select differing levels of generality of inferred rules for different classes.

Case-based critique of complete rule sets.

The previous techniques for case based critique of a rule set concentrated on the evaluation of individual rules in isolation. As mentioned above, for any rule, lists were provided of relevant examples correctly classified, misclassified and not classified. However, while this information is extremely valuable when considering an individual rule, it does not show how the rule set as a whole relates to the example. This is an issue because multiple rules may cover a single example. In consequence, analysis of an individual case might indicate that it is covered by a rule that correctly classifies it, leading the user to believe that it will be handled correctly, while another rule might over-ride the first, resulting in the case's misclassification.

For case-based critique to have broad application it must also be able to support evaluation of a rule set as a whole. We assume that there will be two major concerns with a rule set. First, what cases does it misclassify. Second, what cases does it fail to classify. (In the first situation, the case is assigned the wrong class. In the second, it is assigned no class.)

It turns out to be straight forward to extend the techniques developed for case-based critique of individual rules to cover these situations. Quite simply, two additional windows are maintained, each of which lists one of these types of example case.

Rule application outcome display

Most evaluation of classification rules developed by machine learning systems concentrates on predictive accuracy. However, this is but one of the useful measures by which such a system might be assessed [19]. Different measures of the quality of a classification system will be more appropriate for different contexts. In many contexts predictive accuracy will not be the relevant criterion. If we are to provide useful tools for presenting aggregate summaries of classifier performance on a body of examples, these need to be able to present the type of aggregate summary that is relevant to a particular application.

Rather than have the user select the type of summary required, we have opted to develop a form of summary table that encapsulates all types of summary that are likely to be relevant. An example is presented in Figure 3.

Figure 3: Example aggregate summary table

Conclusion

We have described a number of extensions to existing techniques for integrating knowledge elicitation and machine learning. Of particular novelty is that the key role of machine learning is viewed as identifying clusters of cases to be covered by individual rules rather than formulating precise rules. We have also indicated how the new techniques have been implemented in the proof-of-concept system that we are developing. In addition to these theoretical advances, we have made a number of significant extensions to the implementation, required for various evaluation purposes. These include facilities for generating the rule base in the form of PRL or C code, in addition to the CLIPS and stand-alone Macintosh system that were previously supported. The system's capacity has also been greatly expanded. It is now capable of processing up to 130 attributes.

We believe that the techniques that we have developed allow rapid development of high quality expert systems by naive users. Informal studies that we have conducted support our intuitions. At the time of writing we are planning formal studies to evaluate these intuitions with more rigour.

We believe that the techniques that we are developing have the potential to alleviate the knowledge-acquisition bottle-neck, at least for the class of problems at which it is aimed—those that can be described in attribute-value terms, for which a domain expert has non-trivial insight and for which it is possible to accumulate a suitable set of example cases.

Acknowledgments

This research has been supported by the Australian Research Council.

References

[1] E. Bloedorn and R. S. Michalski, Data-driven constructive induction in AQ17-PRE: A method and experiments, in: *Proceedings of the 1991 IEEE International Conference on Tools for Artificial Intelligence* San Jose, CA, 1991) 30-37.

[2] S. Yip and G. I. Webb, Discriminant attribute finding in classification learning, in: A. Adams and L. Sterling, eds., *AI'92* (World Scientific, Singapore, 1992) 374-379.

[3] C. J. Matheus and L. A. Rendell, Constructive induction on decision trees, in: *IJCAI-89* (Morgan Kaufmann, San Mateo, CA, 1989) 645-650.

[4] S. M. Weiss and N. Indurkha, Reduced complexity in rule induction, in: *IJCAI-91* (Morgan Kaufmann, San Mateo, Ca, 1991) 781-787.

[5] K. Morik, Ed., *Knowledge Representation and Organization in Machine Learning* (Springer-Verlag, New York, 1989).

[6] K. Morik, S. Wrobel, J.-U. Kietz and W. Emde, *Knowledge Acquisition and Machine Learning: Theory, Methods, and Applications* (Academic Press, London, 1993).

[7] R. G. Smith, H. A. Winston, T. M. Mitchell and B. G. Buchanan, Representation and use of explicit justifications for knowledge base refinement, in: *IJCAI-85* (Morgan Kaufmann, San Mateo, Ca, 1985) 673-680.

[8] D. C. Wilkins, Knowledge base refinement using apprenticeship learning techniques, in: *AAAI-88: Proceedings of the Seventh National Conference on Artificial Intelligence* (Morgan Kaufmann, San Mateo, CA, 1988) 646-651.

[9] L. De Raedt, *Interactive Theory Revision* (Academic Press, London, 1992).

[10] G. I. Webb, Man-machine collaboration for knowledge acquisition, in: A. Adams and L. Sterling, eds., *AI'92* (World Scientific, Singapore, 1992) 329-334.

[11] G. I. Webb, Generality is More Significant than Complexity: Toward Alternatives to Occam's Razor, in: C. Zhang, J. Debenham and D. Lukose, eds., *AI'94 – Proceedings of the Seventh Australian Joint Conference on Artificial Intelligence* (World Scientific, Armidale, 1994) 60-67.

[12] P. Compton, et al., 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, New York, 1986).

[14] G. I. Webb, DLGref2: Techniques for inductive knowledge refinement, in: *Proceedings of the IJCAI Workshop W16* Chambery, France, 1993) 236-252.

[15] G. I. Webb, *Is Occam's razor disposable? In support of learning biases based on semantics rather than syntax*, Deakin University School of Computing and Mathematics, Technical Report, TR C95-11 (1995).

[16] C. S. Wallace and D. M. Boulton, An information measure for classification, *Computer Journal* **11** (1968) 185-194.

[17] G. D. Plotkin, A note on inductive generalisation, in: B. Meltzer and D. Mitchie, eds., *Machine Intelligence 5* (Edinburgh University Press, Edinburgh, 1970) 153-163.

[18] 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* (Morgan Kaufmann, Los Altos, 1986), vol. II, 167-191.

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