Beyond Concise and Colorful: Learning Intelligible Rules

Michael J. Pazzani
Subramani Mani
Department of Information and Computer Science
W. Rodman Shankle
Department of Neurology
The University of California
Irvine, CA 92697
pazzani@ics.uci.edu, mani@ics.uci.edu, rshankle@uci.edu

Abstract

A variety of techniques from statistics, signal processing, pattern recognition, machine learning, and neural networks have been proposed to understand data by discovering useful categories. However, research in data mining has not paid attention to the cognitive factors that make learned categories intelligible to human users. We show that one factor that influences the intelligibility of learned models is consistency with existing knowledge and describe a learning algorithm that creates concepts with this goal in mind.

Introduction

Knowledge-discovery in databases is a field whose goal is to extract usable models from a collection of data. Such models are expected to be accurate and are further expected to be intelligible to experts in the field. For example, knowledge acquired through such methods on a medical database might be published in scientific journals or written down as procedures to be followed in a health maintenance organization. While it is important that such knowledge be an accurate summary of the data, it is equally important that the knowledge be comprehensible to experts in the domain. Research in learning comprehensible models has typically equated comprehensible with concise (e.g., Craven, 1996 and Karalic, 1996). Other work on increasing the understandability of learned models concerns the construction of tools for visualizing the results of learning (e.g., The MineSet Tree Visualizer- Kohavi, Sommerfield & Dougherty, 1996). Here we argue that another factor that influences intelligibility of learned rules is being integrated with other knowledge in the domain.

The goal of knowledge-discovery in databases is sometimes viewed as finding "the" model of the data, while in reality there are often many possible models of the data that are not significantly different according to any statistical procedure on the training data. For example, Murphy and Pazzani (1994) used a massively parallel computer to find all decision trees consistent with a set of 20 training examples. A total of over 25,000 trees were found. Many of these trees were very complex. However, on average there were 20 trees with 5 or fewer tests. If it is important that the results of learning be intelligible to people then agreement with known human biases is an additional constraint that may be placed on model selection. Psychological investigation has revealed factors that simplify learning, understanding and communication of category and process information (e.g., Billman & Davila, 1995; Kelley, 1971). In this paper, we will focus on one constraint from psychological investigations: the consistency with prior knowledge (Murphy & Allopenna, 1994; Pazzani, 1991).

This research grew out of analyzing a database collected by the Consortium to Establish a Registry for Alzheimer's Disease (CERAD). A patient database was collected containing data on the dementia status of each patient and the results of two commonly used cognitive tests for dementia screening. The particular problem of interest is to identify patients with early signs of dementia. In previous research (Shankle, Mani, Pazzani, & Smyth, 1997), we have shown that a variety of machine learning and statistical methods can acquire models that have accuracy, specificity and sensitivity that exceed the average practitioner at screening for early stages of dementia. However, it is unlikely that the description of patients with early dementia created by any of the models so far would be widely adopted in practice. The decision procedure implied by some models (e.g., logistic regression) is too complex to follow, while the decision criteria explicitly stated in learned rules or decision trees make little sense to the neurologist or the practitioner since it differs drastically from the current practice.

To understand why the results of current knowledge-discovery algorithms make little sense, it is necessary to

describe how cognitive tests are currently used for screening. In each test, the patient answers questions that assess orientation for time and place, registration, attention, short-term recall, language skills, and drawing ability. For example, the patient is first asked to remember a name and address ("John Brown, 42 Market Street, Chicago") and later asked to recall these items. The patient receives a score for each item in the test. For example, the number of times that the test giver repeats the name and address before the patient is able to repeat it is recorded. An overall score is given to each patient by summing the score on each question. A threshold on the total score is used in practice for screening for dementia.

The score on each question of the test and the patient's age, sex, and years of education were used in our earlier work to predict whether a patient was "normal" or "mildly impaired" by learning algorithms. We showed that such methods would be more effective than a simple threshold on the aggregate score because some questions seemed more important than others. All of the algorithms were more accurate than the simple threshold and none of the methods were substantially more accurate than the others. In such a case, one might prefer to make decisions based upon rules or trees since such representations are easy to follow. However, neither the trees produced by C4.5 (Quinlan, 1993) nor the rules produced by rule learners such as C4.5rules or FOCL (Pazzani & Kibler, 1992) produced rules that would be acceptable in practice. In particular, some items that should be viewed as signs of being impaired are used as signs of being normal and vice versa. Table 1 shows an example of one such rule that was produced by FOCL.

In the remainder of this paper, we first discuss rule learning algorithms using FOCL as an example. We describe an extension to FOCL to prevent it from learning rules that violate the expectations of a domain expert and show that the extension does not hurt the diagnostic value of the learned concepts. We present evidence that one neurologist prefers rules without these violations.

Background: Rule Learners

FOCL is derived from Quinlan's (1991) FOIL system. FOIL is designed to learn a set of clauses that distinguish positive examples of a concept from negative examples. Each clause consists of a conjunction of tests. For example, in the dementia domain a test might check to see whether the patient's age is less than a certain value or whether the age is greater than a certain value.

FOCL follows the same procedure as FOIL to learn a set of clauses. However, it learns a set of clauses for each class (such as normal and impaired) enabling it to also deal with problems that have more than two classes. The clause learning algorithm is run once for each class, treating the examples of that class as positive examples and the examples of all other classes as negative examples. This results in a set of clauses for each class.

Table 1: Rule with questionable tests underlined.

IF the years of education of the patient is > 5
AND the patient does not know the date
AND the patient does not know the name of a nearby street
THEN The patient is NORMAL

OTHERWISE IF the number of repetitions before correctly reciting the address is > 2
AND the age of the patient is > 86
THEN The patient is NORMAL

OTHERWISE IF the years of education of the patient is > 9
AND the mistakes recalling the address is < 2
THEN The patient is NORMAL

OTHERWISE The patient is IMPAIRED

FOCL has an optimization procedure that selects an ordered subset of the original clauses to convert a set of clauses for each class into a single decision list. The algorithm initializes the decision list to a default clause that predicts the most frequent class. Next, it tries to improve upon the current decision list with an operator that replaces the default rule with a learned clause and a new default clause. The impact is calculated of adding each remaining clause to the end of the current decision list and assigning the examples that match no clause to the most frequent class of the unmatched examples. The change that yields the highest impact in accuracy is made and the process is repeated until no change results in an improvement. Typically, only a few clauses are selected by this process resulting in a relatively short decision procedure. Using the same examples to learn the initial set of clauses and to create the ordered decision list can result in overfitting because the learned rules rarely make errors on the learning data. Therefore, we divide the training data into a learning set consisting of 2/3 of the training data for learning clauses and an ordering set consisting of the remaining 1/3 of the training data for creating the decision list. One further detail is needed to understand how FOCL arrives at a decision list using rule optimization. When adding clauses to the decision list, FOCL also has the option to choose a prefix of a learned clause. That is, if a clause such as X&Y&Z was learned, FOCL considers using X or X&Y in addition to X&Y&Z as a clause in the decision list.

Monotonicity Constraints

Some clauses in the learned category descriptions violate the intent of the tests used for screening. In particular, getting some questions right is used as evidence that one is impaired and getting some questions

wrong is used as evidence that one is not impaired. A relatively simple change to FOCL eliminates such tests from consideration. For variables with numeric relationships, the user declares whether the variable has a known monotonic relationship with each class. A monotonic relationship is one in which increasing the value of the variable always increases or decreases the likelihood category membership. When considering tests to add to a clause, the tests that violate these relationships are removed from consideration. For example, when learning a description of the normal patients, FOCL with monotonicity constraints only checks to see if the number of errors recalling the address is less than some number. When learning clauses describing the impaired category, it only tests to see if this variable is above some threshold. These constraints on tests may also be used on Boolean and nominal variables. In this case, the user specifies which values are possibly indicative of membership in a class. For example, a value of true for the variable "knows the date" may be used as a sign for normal, while the value false may be used as a sign for impaired. Table 2 shows an example of a decision list learned with constraints provided by the neurologist working on this project.

If we assume that the constraints are correct, then there are two factors that contribute to a test that violates these constraints being used in a rule. First, while the test appeared best according to an information-based selection procedure, this procedure detected a "spurious correlation" in the data due to sampling biases or noise. Second, there are often several tests that are equally informative or statistically indistinguishable. Under these circumstances, a decision procedure could be found that is both accurate and comprehensible to an expert by eliminating from consideration tests that violate monotonicity constraints.

To test whether a neurologist preferred rules that did not violate constraints, we generated 16 decision lists from constrained and unconstrained FOCL. Each rule was printed on a separate sheet of paper and presented in a random order to a neurologist not involved in this project. We asked the neurologist to rate on a scale of 0-10 "How willing would you be to follow the decision rule in screening for cognitively impaired patients?" We hypothesized that the neurologists would be more willing to use rules that were generated by FOCL when it used monotonicity constraints. The average score of rules generated by FOCL without the monotonicity constraints was 0.25, while the average score of rules generated with the monotonicity constraints was significantly higher at 2.38 $t(15) = 5.09$, $p < .001$.

Violations of the monotonicity constraints

So far, we have assumed that the monotonicity constraints are correct and the learning system does not allow violations of the constraints. Ideally, we would not allow rules that violate the constraints unless violating them results in more accurate rules. Here we describe an extension to FOCL that implements this idea.

Table 2. A rule learned with monotonicity constraints.

IF the years of education of the patient is > 5
AND the mistakes recalling the address is < 2
THEN The patient is NORMAL

OTHERWISE
IF the years of education of the patient is > 11
AND the errors made saying the months backward is < 2
THEN The patient is NORMAL

OTHERWISE
IF the years of education of the patient is > 17
THEN The patient is NORMAL

OTHERWISE The patient is IMPAIRED

The rule optimization algorithm selects from a pool of clauses that contains clauses learned on the training set with constraints and clauses learned from the same training data without constraints. The rule optimization procedure is changed to prefer clauses learned with constraints when the addition of two clauses results in the same increase in accuracy. Initial experimentation with this algorithm revealed that often all of the clauses in the optimized decision list came from one source or the other. This occurs because each set of clauses represents an alternative way of partitioning the training examples into disjunctive sets and clauses from these two different sources usually cannot be recombined to cover the training data. To mitigate this problem, we have further extended FOCL to learn alternative rules from the same training data. We use stochastic search in FOCL to achieve this. Rather then selecting the most informative condition to add to each rule, FOCL selects among the k (3) most informative tests with probability proportional to the informativeness. By repeating the process of learning a set of rules from the training data, several alternative partitions of the data are formed. In the experiment reported below, 5 rule sets are learned without monotonicity constraints and 5 rule sets are learned with monotonicity constraints. These are all entered into the pool of rules for rule optimization.

Table 3 shows the accuracy of C4.5, C4.5rules and FOCL under various conditions with optimized rules on the CERAD data. The accuracy is averaged over 50 trials of dividing the data into a training set of size 210 and a test set of size 105. The test set does not contain any examples from the training set. The four conditions of FOCL reported are 1) no monotonicity constraints, 2) monotonicity constraints, 3) stochastic search and rule optimization selecting from 10 rule sets learned without

monotonicity constraints and 4) stochastic search and rule optimization selecting from 5 rule sets learned without monotonicity constraints and 5 rule sets learned with monotonicity constraints.

Table 3. Accuracy at identifying impaired patients.

Algorithm	Accuracy
C4.5	86.7
C4.5 rules	82.6
FOCL (No constraints)	90.6
FOCL (Monotonicity)	90.7
FOCL (Stochastic)	92.2
FOCL (Combined & Stochastic)	94.5

The first noteworthy result is that FOCL is significantly more accurate than the C4.5 and C4.5 rules at the .01 level using two-tailed t-tests. Second, there is not a substantial or significant difference in accuracy in using the constraints. FOCL is 90.7% accurate when using monotonicity constraint and 90.6% accurate when unconstrained. On average, a decision list formed without constraints contains a total of 4.65 tests and 2.13 violations of the monotonicity constraints. With the constraints, an average of 4.30 tests are used in a decision list, none of which violate the constraints.

The results show that there is an added benefit in selecting from combined multiple sets of rules learned with and without monotonicity constraints. Optimized rules from this source are significantly more accurate (at the .01 level using a paired two-tailed t-test) than optimized rules learned in the same manner without monotonicity constraints. Furthermore, the average number of monotonicity constraint violations is significantly reduced (from 2.06 to 0.75).

Conclusions

We have argued that to be truly useful, the knowledge discovered in databases must both be accurate and comprehensible. We have further argued that one factor that influences the comprehensibility of learned knowledge is the use of conditions as evidence for belonging to some category when prior knowledge indicates that these conditions are evidence that an example does not belong to that category. We have created an enhancement to one algorithm that prevents these conditions from being added to learned models. Finally, we have presented preliminary evidence that experts prefer rules that do not contain violations of prior knowledge.

Acknowledgments

The authors gratefully acknowledge CERAD for collecting and disseminating the database used in this study. This research was funded in part by the Alzheimer's Association Pilot Research Grant, PRG-95-161 and the National Science Foundation grant IRI-9310413.

References

Billman, D. and Davila, D. 1995. Consistency is the hobgoblin of human minds: People care but concept learning models do not. In *Program of the Seventeenth Annual Conference of the Cognitive Science Society*. Erlbaum: Hillsdale, NJ.

Clark, P. and Niblett, T. 1989. The CN2 Induction Algorithm *Machine Learning, 3* 261-284.

Craven, M. W. 1996. Extracting Comprehensible Models from Trained Neural Networks. Ph.D. thesis, Department of Computer Sciences, University of Wisconsin-Madison. (Also appears as UW Technical Report CS-TR-96-1326)

Karalic, A. 1996. Producing More Comprehensible Models While Retaining Their Performance, Information, Statistics and Induction in Science, Melbourne, Australia.

Kelley, H. 1971. Causal schemata and the attribution process. In E. Jones, D. Kanouse, H. Kelley, N. Nisbett, S. Valins, and B. Weiner eds. *Attribution: Perceiving the causes of behavior* (pp 151-174). Morristown, NJ: General Learning Press.

Kohavi, R., Sommerfield, D., and Dougherty J., 1995. Data Mining using MLC++, a Machine Learning Library in C++. IEEE Tools With Artificial Intelligence.

Murphy, G.L. and Allopenna, P.D. 1994. The locus of knowledge effects in concept learning. Journal of Experimental Psychology: Learning, Memory, and Cognition, 19, 203-222.

Murphy, P. and Pazzani, M. 1994. Exploring the Decision Forest: An Empirical Investigation of Occam's Razor in Decision Tree Induction. *Journal of Artificial Intelligence Research* 1 pp. 257-275.

Pazzani, M. 1991. The influence of prior knowledge on concept acquisition: Experimental and computational results. *Journal of Experimental Psychology: Learning, Memory & Cognition, 17, 3,* 416-32.

Pazzani, M. and Kibler, D. 1992. The utility of knowledge in inductive learning. (9):57-94.

Quinlan, J.R. 1990. Learning logical definitions from relations. *Machine Learning, 5,* 239–266.

Quinlan, J. 1993. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, Los Altos, California

Shankle, W.R., Mani, S., Pazzani, M. and Smyth, P. 1997. *Detecting very early stages of dementia from normal aging with machine learning methods.* The Proceedings of the 6th Conference on Artificial Intelligence in Medicine Europe.