Transactions on Information and Communications Technologies vol 6, © 1994 WIT Press, www.witpress.com, ISSN 1743-3517

Generating classification rules from databases

C. Lee

Department of Computer Science and Engineering, University of Connecticut, Storrs, CT 06269, USA

Abstract

Systems for inducing classification rules from databases are valuable tools for assisting in the task of knowledge acquisition for expert systems. In this paper, we introduce an approach for extracting knowledge from databases in the form of inductive rules. We develop an information theoretic measure which is used as a criteria for selecting the rules generated from databases. To reduce the complexity of rule generation, the boundary of the information measure is analyzed and used to prune the search space of hypothesis. The system is implemented and tested on some well known machine learning databases.

1 Introduction

As the hardware and database technology advances, companies have large databases of information, most of which are perhaps lying idle. For example, a hospital might have hundreds of thousands of patient records, or a company might have a database of its customers. The motivation for using rule-based expert systems is well documented and will not be repeated here. It is notoriously difficult to obtain rules directly from human experts [3] [4]. The problem of manual knowledge acquisition for such systems is perhaps their major drawbacks. Furthermore, if the domain requires reasoning under uncertainty, humans are well known to be inconsistent or even contradictory in their description of subjective probabilities [7]. Hence it is quite clear that if our hypothetical company has an existing database of sample data available, a rule induction system would be very useful. As we shall see, the problem can be rendered more general than simply deriving rules for an expert system–in a sense we are involved in a data reduction process, where we want to reduce a large database of information to a small number of rules describing the domain.

2 Information Content of Rules

The format of rules which we will handle in this paper is as follows:

$$\mathrm{A{=}a} \wedge \mathrm{B{=}b} \wedge \cdots \Rightarrow \mathrm{T{=}t}$$

where A, B and T are attributes with a, b and t being values in their respective discrete alphabets. We restrict the right-hand expression to being a single value assignment expression while the left-hand side may be a conjunction of such expressions. The instantaneous information is the information content of the rule given that the left-hand side is true. The critical part is how to define or select a proper measure which can correctly measure the instantaneous information.

ID3, which generates decision trees from data, has been widely used for classification in Quinlan [8]. ID3 uses the following formula as a measure of information.

$$H(T) - H(T|A=a) = \sum_t p(t) \log\left(\frac{1}{p(t)}\right) - \sum_t p(t|a) \log\left(\frac{1}{p(t|a)}\right) \quad . \qquad (1)$$

It calculates the difference between the entropy of a priori distribution and that of a posteriori distribution. However, it is well-known that there is a fundamental problem with this measures. Consider the case of an n-valued variable where a particular value of $T = t$ is one, while all the other values in T's alphabet are zero. In this case, a conditional permutation of these probabilities would be significant, i.e., a rule which predicts the relatively rare event $T = t$. However, the formula (1), because it cannot distinguish between particular events, would yield zero information for such events.

In this paper a new information measure, called Hellinger measure, is used to define the information content of rules. The Hellinger divergence was originally introduced by Beran [1], and is defined as

$$\sqrt{\sum_i \left(\sqrt{p(t_i)} - \sqrt{p(t_i|a)}\right)^2} \qquad (2)$$

where t_i denotes the value of attribute T. It becomes zero if and only if both a priori and a posteriori distributions are identical, and ranges from 0 to $\sqrt{2}$. Unlike other information meaures, this measure is applicable to every possible case of probability distributions. In other words, the Hellinger measure is continuous on every possible combination of a priori and a posteriori values. It can be interpreted as a distance measure where distance corresponds to the amount of divergence between a priori and a posteriori distribution. Therefore, we employ Hellinger measure(H measure) as a measure of divergence, which will be used as the information amount of rules.

3 Properties of H Measure

In terms of the probabilistic rules, let us interpret the event $A = a$ as the concept to be learned and the event(possibly conjunctive) $B = b$ as the hypothesis describing this concept. The information content of the rule is defined as

$$\left[\sqrt{P(a|b)} - \sqrt{P(a)}\right]^2 + \left[\sqrt{1 - P(a|b)} - \sqrt{1 - P(a)}\right]^2 \qquad (3)$$

where $P(a|b)$ means the conditional probability of $A = a$ under the condition $B = b$. Notice that equation (3) has a different form of definition from that of

equation (2). In rule generation, one particular value of class attribute appears in the right hand side of the rule, and thus the probabilities for all other values are included in $1 - P(a)$. In addition, we squared the original form of Hellinger measure because (1) by squaring the original form of Hellinger measure, we could derive a boundary of the H measure, which allows us to reduce the search space of possible inductive rules. (2) the relative accuracy of each rule is not affected by the modified Hellinger measure. (3) the weights between two terms of H measure provides more reasonable trade-off in terms of their value range. This measure can be interpreted as the cross entropy of A with the variable "A conditioned on the event B=b." Cross entropy is well-known as an accuracy measure between two distributions [9].

Another criteria we have to consider is the generality of the rules. The basic idea behind generality is that the often left-hand side occurs for a rule, the more useful the rule becomes. The left-hand side must occur relatively often for a rule to be deemed useful. In this paper, we use $\sqrt{P(b)}$ to represent the probability that the hypothesis will occur and, as such, can be interpreted as the measure of hypothesis generality.

By multiplying the generality with the accuracy of the rules, we have the following term

$$\sqrt{P(b)}\left[\left(\sqrt{P(a|b)}-\sqrt{P(a)}\right)^2+\left(\sqrt{1-P(a|b)}-\sqrt{1-P(a)}\right)^2\right] \qquad (4)$$

which possesses a direct interpretation as a multiplicative measure of the generality and accuracy of a given rule.

The next step is to derive some quantitative bounds on the nature of specialization, which can be used to improve computational performance. The algorithm starts with generating an initial set of rules, followed by specialization of these rules to optimize the rule set. The characteristic of the specialization behavior is critical to the performance of the algorithm. Specialization is the process by which we try to increase a rule's accuracy by adding an extra condition to the rule's left-hand side. The consequent necessary decrease in generality of the rule should be less than an increase in the accuracy to the extent that the overall H measure is increased. The question we pose is as follows: given a particular general rule, what quantitative statements can we make about specializing this rule ? In particular, if we define H_s and H_g as the H measures of the specialized and general rules, respectively, is it possible to find a bound of H_s in terms of H_g ?

Consider that we are given a general rule whose H measure, H_g, is defined as

$$\begin{aligned} H_g &= \sqrt{P(b)}\left[\left(\sqrt{P(a|b)}-\sqrt{P(a)}\right)^2+\left(\sqrt{1-P(a|b)}-\sqrt{1-P(a)}\right)^2\right] \qquad (5)\\ &= \sqrt{P(b)}\left[2-2\sqrt{P(a|b)P(a)}-2\sqrt{(1-P(a|b))(1-P(a))}\right] \end{aligned}$$

We try to calculate the bound of

$$\begin{aligned} H_s &= \sqrt{P(bc)}\left[2-2\sqrt{P(a|bc)P(a)}-2\sqrt{(1-P(a|bc))(1-P(a))}\right] \qquad (6)\\ &= \sqrt{P(c|b)}\sqrt{P(b)}\left[2-2\sqrt{P(a|bc)P(a)}-2\sqrt{(1-P(a|bc))(1-P(a))}\right] \end{aligned}$$

Given no information about C, we can state the following results: Dut to space limitation, the proof of these heuristics are provided in Lee [6].

Heuristics 1 *If the H measure of a specialized rule satisfies the following boundary:*

$$H_s \leq \max\{ \sqrt{P(a|b)}\sqrt{P(b)}\left[2\sqrt{m} - 2\sqrt{P(a)}\right], \\ 2\sqrt{P(b)} - \sqrt{1-P(a|b)}\sqrt{P(b)}\left[2\sqrt{P(a)} + 2\sqrt{1-P(a)}\right]\}$$

where m *represents the number of class in the target attribute, the general rule discontinues specializing.*

As a special case of the Heuristics 1, if the success rate(conditional probability) of general rule becomes 1, the H measure of the specialized rule is always less than or equal to that of general rule.

Heuristics 2 *If the conditional probability of general rule is 1, H measure of specialized rule cannot be greater than that of general rule. Therefore, the general rule discontinues specializing.*

As a consequence of these theorems we note that since the bound of specialized rule is achievable without further information about C, we can decide in advance that the specialized rule cannot be improved with respect to H Measure. The logical consequence of this statement is that it precludes using the bound to discontinue specializing based on the value of H_g alone. Conversely, if $p(a|b)$ is not equal to 1, then with no information at all available about the other variables, there may always exist a more specialized rule whose information content is strictly greater than that of the general rule. However, as we shall see, we could certainly compare the bound with any rules we might already have. In particular, if the bound is less than the information content of the worst rule, then specialization cannot possibly find any better rule. This principle will be the basis for restricting the search space of the system.

4 Rule Generation

We will now define the algorithm and discuss its basis ideas The algorithm takes sample data in the form of discrete attribute vectors and generates a set of K rules, where K is a user-defined parameter. The set of generated rules are the K most informative rules from the data as defined by the H measure. In this sense the algorithm can be described as optimal. The probabilities required for calculating the H measures are estimated directly from the data using standard statistical point estimation techniques. The algorithm proceeds by first finding K rules, calculating their H measures, and then placing these K rules in an ordered list. The smallest H measure, that of the Kth element of the list, is then defined as the running minimum H_*. The critical part of the algorithm is the specialization criterion since it determines how much of the exponentially large hypothesis space actually needs to be explored by the algorithm. The algorithm employs depth-first search over possible left-hand sides, starting with the first-order conditions and specializing from there. The algorithm systematically tries to specialize all first-order rules and terminates when it has determined that no more rules exist which

if success rate of $H_g \neq 1$
then

$$H_s = \max\{\sqrt{P(a|b)}\sqrt{P(b)}\left[2\sqrt{m} - 2\sqrt{P(a)}\right], \\ 2\sqrt{P(b)} - \sqrt{1 - P(a|b)}\sqrt{P(b)}\left[2\sqrt{P(a)} + 2\sqrt{1 - P(a)}\right]\}$$

if $H_s \leq H_*$ **then** cease to specialize; /* Heuristics 1 */
else
cease to specialize; /* Heuristics 2 */

Figure 1: Algorithm for specialization

can be specialized to achieve a higher H measure than H_*. The decision whether to continue specializing or to back-up on the depth-first search is determined by the algorithm in Figure 1.

Due to the inductive nature of the rules the system generates, there exists the possibility that some rules are contradictory with each other. By providing a way to decide which class the new instance belongs, we can easily turn the current rule induction system into a classification system. When a new instance is given, we first try to classify the instance using the inductive rules generated from the system. Because of the presence of uncertainty, new instance might match more than one class, thus, there is a need to decide which specific class it should be assigned to. For this purpose, the H measure of each inductive rule can be considered as the weight of the rules. For a given new instance, the system selects the rules which the new instance can fire, and, among the rules selected, if a general rule subsumes special rules, the special will be deleted. In words, for each rule matched with the new instance, the H measures are accumulated based on the class values, and the class value with the largest H measure collected will be selected as the final class value.

4.1 Missing Values

The presence of incompletely described instances complicates learning. Rule generation from incomplete data requires an effective method for handling missing attribute values. In this paper, we treat "unknown" as a new possible value for each attribute and deal with it in the same way as other values. When calculating the H measure of each inductive rule with the presence of missing values, the system checks whether each instance satisfies the left hand side of the rule. If the instance matches the left hand side of the rule, the system updates the corresponding conditional probability of the rule, $P(a|b)$ of equation 4. The second important part is how to process the missing values in new instances. For a new instance, the system looks for rules which the new instance matches. As we mentioned earlier, the system handles missing value as another possible value. When a value of an attribute A is missing, that instance cannot match any of the rules which have attribute A as part of their left hand side conditions.

Rules		H
PL < 24.5	⇒ S=Setosa	0.4886
PW < 8.0	⇒ S=Setosa	0.4886
51.5 < PL	⇒ S=Virginica	0.4029
18.5 < PW	⇒ S=Virginica	0.4029
24.5 < PL < 44.5	⇒ S=Versicolor	0.3721
8.0 < PW < 13.5	⇒ S=Versicolor	0.3656
54.5 < SL and 33.5 < SW < 37.5	⇒ S=Setosa	0.3090
SL < 70.5	⇒ S=Virginica	0.2394
29.5 < SW < 33.5 and 13.5 < PW < 15.5	⇒ S=Versicolor	0.2185
SL < 54.5	⇒ S=Setosa	0.2080
45.5 < PL < 47.5	⇒ S=Versicolor	0.1954
SL < 54.5 and 29.5 < SW < 33.5	⇒ S=Setosa	0.1545
SL < 54.5 and 37.5 < SW	⇒ S=Setosa	0.1201
44.5 < PL < 45.5 and 13.5 < PW < 15.5	⇒ S=Versicolor	0.1197
57.5 < SL < 65.5 and 44.5 < PL < 45.5	⇒ S=Versicolor	0.1197

Figure 2: Rules from iris data

5 Experimental Results

We have applied the system to two typical databases: iris flower data set and Monk robot data, obtained from the University of California Irvine machine learning database repository. For each data set, the entire data set is read and then the system generates inductive rules. The results are analyzed in the following.

Iris Flower Data

Iris flower database is perhaps the best known database to be found in the classification literature (e.g., Breiman [2]). Fisher's paper is a classic in the field and is referenced frequently to this day. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. The four attributes are sepal length(SL), sepal width(SW), petal length(PL), and petal width(PW), and their ranges are 43-79, 20-44, 10-69, and 1-24 respectively. For simplicity, the values of iris database are discretized into seven intervals before the system reads this database. We have applied the context-sensitive discretization method described in Lee and Shin [5].

The system generates 110 rules by reading the iris data set, and among them the 15 most informative rules are selected as shown in Figure 2. In essence, the rules in Figure 2 effectively summarize the hidden characteristics of iris data and these rules can be used to classify new iris instances for classification. By selecting 25 rules, the system could classify all 150 instances, and classified 148 instances correctly which results in 98.6% accuracy. To the best of our knowledge, this classification accuracy is the best result ever known for the iris data set. Figure 3 shows the number of instances the system can classify using the selected rules.

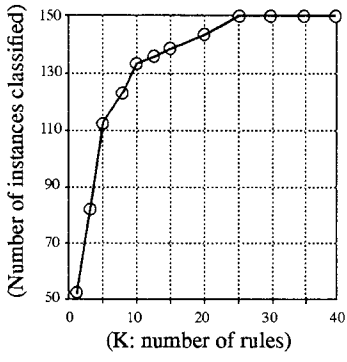

Figure 3: Number of instances classified by the selected rules

Monk Robot Data

The Monk problems take place in an artificial robot domain where robots are described by six different attributes: (1) head_shape: round, square, octagon (2) body_shape: round, square, octagon (3) is_smiling: yes, no (4) holding: sword, balloon, flag (5) jacket_color: red, yellow, green, blue (6) has_tie: yes, no (7) class: 0, 1. The learning task is a binary classification task. Each problem is given by a logical description of a class. There are 124 training instances and 432 test instances. The testing examples are all possible examples(216 positive and 216 negative). Among the rules generated from the system, the top 15 most informative rules are shown in Figure 4. As the author of the Monk data set mentioned, Monk data contains the following implicit rules: (jacket_color = red) or (head_shape = body_shape). We can see that, in Figure 4, these rules are generated from the system as the top 4 rules. By selecting 80 rules, the system could classify all instances, and these rules are used to classify new instances for classification. The system classified 401 correctly out of 432 instances which results in 92.8% accuracy.

6 Conclusion

In this paper we have introduced a method of generating inductive classification rules from databases. We developed an information theoretic measure which becomes the criteria for selecting and sorting inductive rules generated. The boundary of the H measure is analyzed and two heuristics are used to reduce the computational complexity of the system. The algorithm is applied to a couple of famous machine learning databases. Missing values can be handled by considering them as separate categories. The resulting rules generated from the data sets show how the system describes the hidden pattern of data sets effectively.

Rule		H
jacket=red	⇒ class=1	0.2849
head=octagon, body=octagon	⇒ class=1	0.2181
head=square, body=square	⇒ class=1	0.2049
head=round, body=round	⇒ class=1	0.1587
head=round, body=octagon, jacket=blue	⇒ class=0	0.1496
head=octagon, holding=sword, tie=yes	⇒ class=1	0.1496
head=round, smiling=y, jacket=blue	⇒ class=0	0.1400
head=round, jacket=blue, tie=no	⇒ class=0	0.1400
head=square, body=round, smiling=n	⇒ class=0	0.1400
head=round, body=square, smiling=n	⇒ class=0	0.1400
head=square, body=round, jacket=blue	⇒ class=0	0.1296
head=round, body=octagon, smiling=y, tie=yes	⇒ class=0	0.1296
head=round, body=square, jacket=green	⇒ class=0	0.1296
head=octagon, smiling=y, tie=yes	⇒ class=1	0.1296
head=round, body=octagon, jacket=yellow	⇒ class=0	0.1296

Figure 4: Rules from Monk data

References

1. Beran, R. J. Minimum Hellinger Distances for Parametric Models, *Ann. Statistics*, Vol. 5, pp. 445-463, 1977.

2. Breiman, L., Friedman, J., Olshen, R. & Stone, C. *Classification and Regression Trees*, Belmont: Wadsworth, 1984.

3. Hart, A. *Knowledge Acquisition for Expert Systems*, New York: McGraw Hill, 1986.

4. Johnson, N. E. Mediating Representations in Knowledge Elicitation, *Proceedings of the First European Workshop on Knowledge Acquisition for Knowledge-Based Systems*, Reading, England, 1987.

5. Lee, C. H. & Shin, D. G. A Context-sensitive Discretization Method for Classification Learning, *Proceedings of the 11th European Conference on Artificial Intelligence*, Amsterdam, Netherland, 1994.

6. Lee, C. H. *A Hybrid Approach for Classification Learning in Databases*, Ph.D. Thesis, Department of Computer Science and Engineering, University of Connecticut, 1994.

7. Kahneman, D., Slovic, P. & Tversky, A. *Judgement under Uncertainty: Heuristics and Biases*, Cambridge, England: Cambridge University, 1982.

8. Quinlan, J. R. Induction of decision trees, *Machine Learning*, 1, pp. 81-106, 1986.

9. Shore, J. E. & Johnson, R. W. Axiomatic Derivation of the Principle of Maximum Entropy and the Principle of Minimum Cross-Entropy, *IEEE Transactions on Information Theory*, Vol. 26, No. 1, 1980.

10. P. Smyth and R. M. Goodman. Rule Induction Using Information Theory, in *Knowledge Discovery in Databases*, G. P. Shapiro and W. J. Frawley, editor, AAAI Press, pp. 159-176, 1991.