International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:7, 2007

Mining of Interesting Prediction Rules with Uniform Two-Level Genetic Algorithm

Bilal Alatas and Ahmet Arslan

***Abstract*—** The main goal of data mining is to extract accurate, comprehensible and interesting knowledge from databases that may be considered as large search spaces. In this paper, a new, efficient type of genetic algorithm (GA) called uniform two-level GA is proposed as a search strategy to discover truly interesting, high-level prediction rules, a difficult problem and relatively little researched, rather than discovering classification knowledge as usual in the literatures. The proposed method uses the advantage of uniform population method and addresses the task of generalized rule induction that can be regarded as a generalization of the task of classification. Although the task of generalized rule induction requires a lot of computations, which is usually not satisfied with the normal algorithms, it was demonstrated that this method increased the performance of GAs and rapidly found interesting rules.

***Keywords*—** Classification Rule Mining, Data Mining, Genetic Algorithms.

I. Introduction

DATA mining (DM) consists of the discovery of highly accurate, comprehensible and interesting (novel) knowledge from large databases. There are several kinds of tasks of DM depending mainly on the application domain and the user interest. The task of classification task consists of supervised learning methods that induce a classification model from a database. However, in many classification algorithms, the emphasis is discovery of accurate knowledge as measured e.g. by the classification error rate. In this paper, knowledge accuracy is secondary concern and the emphasis is discovering novel, interesting (surprising), comprehensible knowledge.

As demonstrated in various application domains, GAs have proved to be an appealing alternative to classical search algorithms for exploring a large search space. Besides their robustness and less likely to getting stuck in local optima, they have tendency to cope better with attribute interaction. Moreover, they are highly parallel in nature and therefore attractive to parallel and distributed implementations.

This paper proposes a new, efficient type of GA, called uniform two-level GA, to discover interesting rules for the task of generalized rule induction where different rules can predict different goal attributes. This task can be regarded as a generalization of the very well known classification task, where all rules predict the same goal attribute. The two key issues in the proposed approach are the use of uniform population [11-12], which distributes the initial population in the feasible region uniformly and the new type of GA, two-level GA, which uses an island model for initial population and distributes initial population on different islands methodically.

This paper is organized as follows. Section 2 briefly describes the related works about interestingness of the rules. The basic characteristics of the task of generalized rule induction and advantages of using GAs for this task from a DM viewpoint is also described Section 3. Section 4 is the detailed description of proposed method. Section 5 briefly describes the data set used in the experiments. Section 6 discusses the experimental results. Finally, section 7 concludes the paper.

B. Alataş is with Department of Computer Engineering, Fırat University, 23119, Elazığ / Turkey (phone: +90 424 237 00 00 / 5293; fax: +90 424 218 19 07); e-mail: balatas@firat.edu.tr).

A. Arslan is with Department of Computer Engineering, Selcuk University, Konya / Turkey (e-mail: ahmetarslan@selcuk.edu.tr).

II. Related works about rule interestingness

Recently, several researchers have presented different viewpoints on the rule interestingness. In [1] the need for a better grasp on the concept of interestingness for DM with an example from marketing is demonstrated. Applying a traditional apriori association algorithm to the analysis of 87,437 records of consumer purchase data, over 40,000 association rules were generated, “many of which were irrelevant or obvious.” Identifying the important and actionable discoveries from amongst these 40,000 “nuggets” is itself a key task for DM.

The concept of *interestingness* is difficult to formalize and varies considerably across different domains. A growing literature in DM is beginning to address the question. Early work attempted to identify objective measures of interestingness, and the confidence and support measures used in association algorithms are examples of objective measures. One of the earliest efforts to address the explosion of discoveries by identifying interestingness was through the use of rule templates with attribute hierarchies and visualization [2]. In [3], interestingness measures are partitioned into objective and subjective measures, and further partition subjective measures into those that capture unexpectedness and those that capture actionability.

Many authors have focused on capturing unexpectedness as a useful measure, particularly in the context of discovering associations [4] and classifications [5]. In [6], an

unexpectedness algorithm based on logical contradiction has been developed in the context of expectations or beliefs formally expressed for an application domain.

Capturing actionability is a difficult and less studied proposition. In [7], the concept of payoff as a measure of interestingness, where they attempt to capture the expected payoff from the actions that follow from their discoveries (deviations) is discussed. There is little other work specifically addressing actionability.

In [8-9], objective measures of rule interestingness have been discussed and the degree is calculated by an information-theoretical measure. Briefly, while some of the researchers discuss objective measures, the others propose subjective measures to evaluate rule interestingness. This paper uses the ideas from [10] to evaluate rule interestingness in an objective manner.

III. Genetic Algorithms and Generalized Rule Induction

Classification systems are useful techniques in DM, which are supervised learning methods that induce a classification model from a database. They have been studied extensively in the fields of statistics, pattern recognition, decision theory, machine learning literature, neural network, etc [13]. The database, training set, is composed by a set of attributes, or features, of records and each tuple has a known class, or label, associated with it. The goal is to induce a set of classification rules for each class using the attributes available in the database. Such rules are used to classify future records according to the value of their attributes.

The discovered knowledge is usually presented in the form of IF-THEN prediction rules because this method presents a high-level, symbolic knowledge presentation and contributes the comprehensibility of the discovered knowledge. The discovered rules can be evaluated according to several criteria, such as the degree of confidence in the prediction, classification accuracy rate on unknown-class instances, comprehensibility, etc. Another crucial criterion in the spirit of DM is the interestingness of the rules.

Generalized rule induction is a DM task that can be seen as a generalization of the task of classification. In classification, the goal is to predict the value of a special goal attribute, given the values of other predicting attributes. Hence all rules have the same attribute in their consequent. Generalized rule induction addressed by the proposed GA consists of predicting the values of a small number of goal attributes given the values of other predicting attributes. Unlike classification, there is more than one goal attribute to be predicted. Hence, different rules can have different attributes in their consequent.

Briefly, task of generalized rule induction is a natural generalization of the task of classification. In the proposed approach, a small set of goal attributes that the user is interested in predicting is specified. Just a few user-selected attributes can occur either in the antecedent or consequent of the rule.

DM techniques have significant problems. First, underlying functions are often non-linear, with variables that have different relationships in different points of space. Second, data can have noise or errors in recording attribute value or classifying the instance. Third, the scale of DM problems is often enormous because of large number of dimensions and mixed data types.

GAs find many applications and are used to solve complex problem. They have also been used in classification [14] and generalized rule induction [15]. Because GAs are robust and they approach uniformly to large number of different classes of problems. If the solution for a given problem exists, the GA with proper coding, operators and fitness function will find it. This is an obvious advantage over other methods such as regression models that can only be used in specific cases. Such generality is desirable in DM where the search space is complex and contain noise.

GAs require no knowledge about the search space and discontinuities present on the search space have little effect on overall search process. They perform very well for large-scale search problem. Such advantages are desired properties for the task of generalized rule induction that has much larger search space than that of the task of classification has.

GAs perform a global search using a population of candidate solutions, rather than using a single candidate solution at a time. In contrast most DM methods are based on the rule induction paradigm, where the algorithm usually performs a kind of local search. Related motivation for using GAs is their success in coping with attribute interaction. GAs do not select one attribute at a time and do not evaluate a partially constructed candidate rule, therefore they are not sensitive to attribute interaction problems. Another important feature of GAs is their easily adaptation to changing condition in the system.

IV. The Proposed Method

A. Uniform Population (UP) Method

All genetic solutions for any optimization problem have been done by means of the creating initial population randomly. However this kind of method has some drawbacks. Initial population may be created in the infeasible region, or all the chromosomes in population may be in the nearest neighborhood of each other's and faraway from solution, or search of solution may get a local solution and we cannot get rid of this local solution. In this study, we used a method to create initial population called uniform population method [11-12]. This method distributes initial population over chromosomes space uniformly and then solution point has at least one chromosome of d-neighborhood where d=hamming distance between solution point chromosome and the nearest chromosome to solution chromosome. However, random initial method will not guarantee this case.

For binary encoding, let $x = (x_1, x_2, ..., x_n)$ be a row vector (individual or chromosome) and $x_i \in \{0,1\}$, $1 \le i \le n$. There is a parameter, r, for this method. If r=1, then initially, a

chromosome is randomly created and then, inversion of this chromosome is also selected as another chromosome. If r=2, then randomly created chromosome is divided into two equal parts: First, the inversion of the first part is taken and this yields another chromosome. Taking inversion of the second part will yield another chromosome, and inversion of all genes of randomly generated chromosome is also another chromosome. Therefore, three extra chromosomes are obtained from randomly created chromosome. All these chromosomes are related to each other. For example, a population of size 4m is created from m randomly created chromosomes (m is a positive integer) in case of r=2. Let x be a randomly created chromosome, and y, z, and t derived chromosomes from x for r=2. Then,

$$a = (x_1, x_2, ..., x_n)$$
$$b = (\bar{x}_1, \bar{x}_2, ..., \bar{x}_n) \quad (1)$$
$$c = (\bar{x}_1, \bar{x}_2, ...\bar{x}_{\lfloor \frac{n}{2} \rfloor}, x_{\lfloor \frac{n}{2} \rfloor + 1}, ..., x_n)$$
$$d = (x_1, x_2, ..., x_{\lfloor \frac{n}{2} \rfloor}, \bar{x}_{\lfloor \frac{n}{2} \rfloor + 1}, ..., \bar{x}_n)$$

If the parameter for each randomly generated chromosome is selected as r, then the number of derived chromosomes from a randomly generated chromosome is 2^r-1. Thus, the number of chromosomes in the initial population will be (the number of randomly generated chromosomes is p)

$$(2^r\text{-}1) \times m + m = m \times 2^r \quad (2)$$

This method can also be used for other types of encoding with a small modification [12].

B. Uniform Population with Island Model

In the uniform population, the chromosome generated at random is divided into parts related to the value of r and then new chromosomes are generated by methodically complement of the parts of the chromosome. In the island model, whereas chromosomes generated at random are placed in one island, complements of these chromosomes are placed in another island. Chromosomes generated by complementing the right parts are placed in different islands, and by this operation, 2^r islands are generated. Figure 1 shows the chromosomes and addition of these chromosomes into islands.

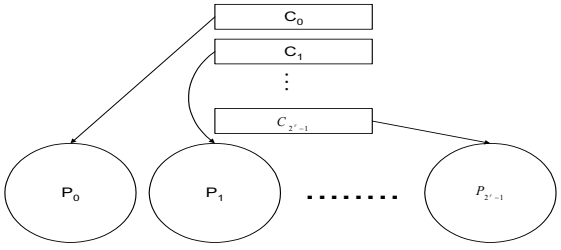

FIGURE 1
ADDITION OF THE CHROMOSOMES INTO ISLANDS

Let C_0 be the chromosome generated at random. C_1, C_2, ..., C_{2^r-1} chromosomes are generated from C_0. These chromosomes, as shown in Figure 1, are added to the islands that have the same index with themselves. In this figure, situation of one chromosome generated at random is shown. The situation is the same when more chromosomes are generated at random. It is important to see that when all C_0 chromosomes are different from each other's, the chromosomes added to islands will be different from each other's. If all C_0 chromosomes are same, only the chromosomes in each island will be same and there will be no similarity between the islands. This characteristic of the island model is better than that of random population method has, because the population in the islands may be the same because of the randomness.

After finding the best chromosomes on different islands, they are combined in the first island and another genetic search is fulfilled to find all rules.

An important characteristic of the island model is that, it has a very convenient structure for parallel or distributed architecture. As emphasized before, because there is no chromosome similarity between islands in the beginning, each machine will solve the problem with different population.

C. Encoding

A chromosome corresponds to a single rule of the form "*if A then C*", where *A* is the antecedent of the rule consisting of a conjunction conditions on the values of at most *n-1* predicting attributes, where *n* is the number of attributes being mined. *C* is the consequent of the rule consisting of a single condition specifying the value predicted for a given goal attribute. Then a chromosome is composed of n genes, where each gene corresponds to a condition containing one attribute. Each *m-th* gene is partitioned into two fields: flag (f_i), and value (v_i) as shown in Figure 2.

The antecedent of the rule is encoded as a variable-length list of rule conditions. If an attribute is not present in the antecedent of the rule, the value of its flag is 0. The value field involves one of the values belonging to the domain of the attribute A_i.

$Gene_1$		$Gene_2$		...	...	$Gene_n$	
f_1	v_1	f_2	v_2	...	...	f_n	v_n

FIGURE 2
CHROMOSOME REPRESENTATION

Each antecedent condition consists of the form $A_i = V_{ij}$, where A_i is the *i-th* predicting attribute and V_{ij} is the *j-th* value of the *i-th* predicting attributes' domain. A chromosome's consequent consists of the form $G_k = V_{kl}$, where G_k is the *k-th* goal attribute and V_{kl}, is the *l-th* value of the *k-th* goal attribute. Note that although each rule predicts the value of a single goal attribute, in a given population different rules can

predict the value of different goal attributes.

The antecedent of the rule is generated when the initial uniform population is created and is thereafter subject to the action of crossover and mutation as will be seen below. However, the consequent of the rule is formed in a special way that maximizes the fitness of a chromosome, because it is very important for the quality of the rule represented by the chromosome. The generation of the consequent of the rule is delayed until fitness computation time, when antecedent of the rule is formed in such a way that maximizes the predictive accuracy of the rule.

D. Fitness Function

The fitness function consists of two terms. The first one measures the degree of interestingness of the rule in an objective (data-driven, domain independent), while the other measures its predictive accuracy.

The term of interestingness consists of two parts. One of them is the interestingness of the antecedent of the rule and the other is interestingness of the consequent of the rule. The degree of the interestingness of the antecedent of the rule is computed by information theoretical measure that is a normalized version of the measure proposed by [10]. The degree of the interestingness of the antecedent of the rule (*AI*) is given by:

$$\mathrm{AI} = 1 - \left[\frac{\sum_{i=1}^{n} InfoGain\ (A_i)}{\frac{n}{\log_2 (|\ dom\ (G_k\ |)}} \right] \quad (3)$$

Here, n is the number of attributes in the antecedent; $(|\ dom(G_k\ |)$ is the number of possible values of the goal attribute G_k occurring in the consequent. The log term is used to normalize the value of *AI* so that this measure takes on a value between 0-1. The *InfoGain* is given by:

$$\mathrm{InfoGain}\,(\mathrm{A_i}) = \mathrm{Info}(\mathrm{G_k})\ \ \mathrm{Info}(\mathrm{G_k} \mid \mathrm{A_i}) \quad (4)$$

where

$$\mathrm{Info}(\mathrm{G_k}) = -\sum_{j=1}^{m_k} (\Pr(V_{kj}) \log_2 (\Pr(V_{kj}))) \quad (5)$$

$$\mathrm{Info}(\mathrm{G_k} \mid \mathrm{A_i}) = \sum_{z=1}^{n_i} \left(\Pr(V_{iz}) \left(-\sum_{j=1}^{m_k} \Pr(V_{kj} \mid V_{iz}) \log_2 (\Pr(V_{kj} \mid V_{iz})) \right) \right) \quad (6)$$

Here, m_k is number of possible values of the goal attribute G_k, n_i is the number of possible values of the attribute A_i. *Pr(X)* states the probability of *X* and *Pr(X|Y)* states the conditional probability of *X* given *Y*.

All other things being equal [10] argues that rules whose antecedent contain attributes with low information gain are more interesting than rules whose antecedent contain attributes with high information gain.

The computation of the degree of interestingness of the consequent of the rule associated with the chromosome is based on the following rationale [10]. The larger the relative frequency of the value being predicted by the consequent of the rule in the training set, the less interesting it is. That is to say, the rarer a value of a goal attribute, the more interesting a rule predicting it is. That is why, the formula for measuring degree of the interestingness of the consequent of the rule (*CI*) is:

$$\mathrm{CI} = (1 - \Pr(\mathrm{G_{kl}}))^{1/\beta} \quad (7)$$

where $Pr(G_k)$ is the relative frequency of the goal attribute G_{kl}, and β is a user-defined parameter empirically set to 2 in the experiments.

The second term of the fitness function is used for the predictive accuracy (PA) of the rule, and it is given by:

$$\mathrm{PA} = (\mathrm{X} - 1/2) / \mathrm{Y} \quad (8)$$

X is the number of instances that satisfy both the antecedent and consequent of the rule; *Y* is the total number of instances that satisfy the antecedent of the rule. The term ½ is used to penalize rules covering few training instances.

Finally the fitness function is given by:

$$\mathrm{Fitness} = \frac{w_1 . \frac{AI + CI}{2} + w_2 . PA}{w_1 + w_2} \quad (9)$$

where w_1 and w_2 are user-defined weights and set to 1 and 2 respectively.

E. Genetic Operators

The current version of the system handles only categorical attributes and used operators are suitable for this situation.

Tournament selection and uniform crossover with probability=100% are used, and elitist strategy is applied. For crossover operator, the probability of genes at division points of uniform population method to be exchanged is zero.

The mutation operator transforms the value of an attribute into another value belonging to the domain of that attribute. The mutation probability is 0.05.

Besides crossover and mutation there are two different operators called insert and remove operator that directly control the size of the rules being evolved for comprehensibility. These operators insert or remove, respectively, a condition in the antecedent of the rule at random. Note that crossover, mutation and these operators act in the antecedent of the rule. Once these operators have been applied and antecedent of the rule is formed, the best consequent is chosen.

V. THE USED DATA SET

The data set used in the algorithm was obtained from the UCI (University of California at Irvine) – Machine Learning Repository. The used data set is Nursery database. This database was derived from a hierarchical decision model originally developed to rank applications for nursery schools. It was used during several years in 1980's when there was excessive enrollment to these schools in Ljubljana, Slovenia, and the rejected applications frequently needed an objective explanation. The final decision depended on three sub problems: occupation of parents and child's nursery, family structure and financial standing, and social and health picture of the family. The model was developed within expert system shell for decision-making DEX [16].

In this work, this database is chosen for the task of generalized rule induction. It contains more than one potential goal attributes that are suitable for this task.

This database contains 12960 instances each of them representing an application for admission in the public school system. Each instance has 9 attributes. All attributes are categorical. The attributes names are as follows: parents, health, form, children, finance, housing, social, has_nurs, and recommendation. Finance, social, and recommendation with respectively 2, 3, and 5 values are chosen as the potential goal attributes.

VI. EXPERIMENTAL RESULTS

In the task of classification, the accuracy rate, i.e. the ratio of the number of the correctly classified test instances over the total number of test instances has been used as a measure of predictive accuracy. In the task of generalized rule induction, in the sense addressed in this paper, the goal is not to classify the whole test set. Rather the goal is to discover a few interesting rules. These discovered rules are valuable they do not cover the whole data set. The value of the discovered rules depends on their predictive accuracy on the part of the test set covered by those rules, but not on the test set as a whole.

The experiment consists of using the full data set to discover the interesting rules. GA run consists of 48 chromosomes with 4 islands each of which operates with 12 chromosomes. GA run was terminated after 40 generations. Other parameters set was defined in Section 4.

The final rules discovered from the data set are shown in Table 1. The table shows the best rule for each possible consequent. For each rule, the degree of interest of antecedent, *AI*, the degree of interest of consequent, *CI*, the number of instances covered by the antecedent, *Y*, and the number of correctly predicted instances, *X*, are also shown in the table. Although the obtained results are similar to the results obtained in [15], the proposed GA takes less iteration and rapidly finds interesting rules.

VII. CONCLUSION AND FUTURE WORKS

A GA-based interesting rule mining that uses a novel population generation and island model has been presented for the task of generalized rule induction. With the uniform two-level GA using suitable chromosome encoding and fitness function, the obtained results are promising since the discovered rules by the new GA are both accurate and interesting.

The proposed GA was designed to effectively discover a few interesting, high-level rules rather than to discover a large set of rules. Because large set of rules may be accurate but it is not necessarily interesting and, it is very easy for the user to understand the concise set of interesting, comprehensible rules.

Although the task of generalized rule induction requires a lot of computations, which is usually not satisfied with the normal algorithms, it was demonstrated that this method has coped with the problems of GAs such as divergence of genetic search process and remaining stuck on local solution of genetic search, and rapidly found interesting rules. The method has a very convenient structure for parallel or distributed architecture and we plan a parallel implementation of this method to extend to cope with continuous attributes with more elaborated experiments by using optimized parameters.

REFERENCES

[1] B. Padmanabhan and A. A. Tuzhilin, "Belief-driven method for discovering unexpected patterns", in *Proc. KDD98 4th International Conference on Knowledge Discovery and Data Mining*, AAAI Press, 1998.

[2] M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A. I. Verkamo, "Finding interesting rules from large sets of discovered association rules", in N. R. Adam, B. K. Bhargava and Y. Yesha (eds), *Proc. 3rd International Conference on Information and Knowledge Managemen*t, ACM Press, pp. 401–407, 1994.

[3] A. Silbershatz and A. Tuzhilin, "What makes patterns interesting in knowledge discovery systems", *IEEE Transactions on Knowledge and Data Engineering*, 8(6), pp. 970–974, 1996.

[4] A. Silbershatz and A. Tuzhilin, "On subjective measures of interestingness in knowledge discovery", in U. M. Fayyad and R. Uthurusamy (eds), KDD95: *Proc. 1rst International Conference on Knowledge Discovery and Data Mining*, AAAI Press, pp. 275–281, 1995.

[5] B. Liu, W. Hsu, and S. Chen, "Using general impressions to analyse discovered classification rules", in *Proc. KDD97 3rd International Conference on Knowledge Discovery and Data Mining*, AAAI Press, pp. 31– 36, 1997.

[6] B. Padmanabhan and A. Tuzhilin, "A belief-driven method for discovering unexpected patterns", in *Proc. KDD98 4th International Conference on Knowledge Discovery and Data Mining*, AAAI Press, 1998.

[7] C. J. G. Piatetsky-hapiro and D. McNeill, "Selecting and reporting what is interesting", in U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy (eds), *Advances in Knowledge Discovery and Data Mining*, AAAI Press, pp. 465–515, 1996.

[8] E. Noda, A.A. Freitas, and H. S. Lopes, "Discovering interesting prediction rules with a genetic algorithm", in *Proc. Congress on Evolutionary Computation (CEC-99)*, 1322-1329, Washington D.C., USA, 1999.

[9] E. Suzuki and Y. Kodratoff, "Discovery of surprising exception rules based on intensity of implication", in *Proc. 2nd European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD-98)*, Lecture Notes in Artificial Intelligence, 1510, pp. 10-18, 1998.

[10] A. A. Freitas, "On objective measures of rule surprisingness", in *Proc. 2nd European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD-98)*, Lecture Notes in Artificial Intelligence, 1510, pp.1-9, 1998.

[11] A. Karcı and A. Çınar, "Comparison of uniform distributed initial population method and random initial population method in genetic search", in *Proc. 15th International Symposium on Computer and Information Sciences*, pp. 159-166, Istanbul / Turkey, 2000.
[12] A. Karcı and A. Arslan, "Uniform population in genetic algorithms", *Journal of Electrical and Electronics*, Istanbul Unv, pp. 495-504, vol:2, no:2. 2002.
[13] J. Han and M. Kamber, "*Data mining: concepts and techniques*", Morgan Kaufmann Publishers, 2001.
[14] K. K. Gündoğan, B. Alataş, A. Karci, Y. Tatar, "Comprehensible classification rule mining with two-level genetic algorithm", *in Proc. 2nd FAE International Symposium*, pp. 373-377, Cyprus, 2002.
[15] A. A. Freitas, "A genetic algorithm for generalized rule induction", In: R. Roy et al. Advances in Soft Computing - Engineering Design and Manufacturing, 340-353, (*Proc. WSC3, 3rd On-Line World Conference on Soft Computing*, hosted on the Internet) Springer-Verlag, 1999.
[16] M. Bohanec, V. Rajkovic, "Expert system for decision making", *Sistemica* 1(1), pp. 145-157, 1990.

TABLE I
RESULTS FROM THE NURSERY DATA SET

<Goal, Value>	*Discovered Rule*	*AI*	*CI*	***X***	***Y***
<finance, convenient>	If (has_nurs = very_crit) and (children = more) and (health = recommended) and (class = priority) then (finance = convenient)	0.998888	0.706733	14	14
<finance, inconv>	If (has_nurs = very_crit) and (housing = convenient) and (social = slightly_prob) and (health = recommended) and (class = spec_prior) then (finance = inconv)	0.999111	0.707481	20	20
<social, non_prob>	If (form = complete) and (housing = critical) and (class = very_recom) then (social = non_prob)	0.996424	0.816497	10	10
<social, slightly_prob>	If (form = completed) then (social = slightly_prob)	1.0	0.816497	720	2160
<social, problematic>	If (parents = usual) and (has_nurs = critical) and (health = recommended) and (class = spec_prior) then (social = problematic)	0.997318	0.816497	63	63
<recommendation, not_recom>	If (parents = usual) and (form = foster) and (housing = less_conv) and (finance = inconv) and (health = non_recom) then (class = not_recom)	0.949673	0.816497	120	120
<recommendation, recommended>	If (children = 1) then (class = recommended)	0.997195	0.999882	2	2160
<recommendation, very_recom>	If (parents = pretentious) and (has_nurs = less_prop) and (housing = convenient) and (finance = convenient) and (social = slightly_prob) and (health = recommended) then (class = very_recom)	0.943364	0.980528	16	16
<recommendation, priority>	If (parents = pretentious) and (has_nurs = less_prop) and (form = more) and (health = priority) then (class = priority)	0.934211	0.774445	72	72
<recommendation, spec_prior>	If (parents = usual) and (has_nurs = very_crit) and (form = more) and (finance = inconv) and (health = priority) then (class = spec_prior)	0.934211	0.878845	72	72