# A Genetic Algorithm for Discovering Interesting Fuzzy Prediction Rules: applications to science and technology data

**Wesley Romão**
UEM, DIN-CTC
Av. Colombo, 5790
Maringá – PR.
87020-900 - Brazil
wesley@din.uem.br
http://www.din.uem.br/~wesley
(55) (44) 263-2479

**Alex A. Freitas**
PUC-PR, PPGIA-CCET
Rua Imaculada Conceição, 1155
Curitiba – PR.
80215-901 – Brazil
alex@ppgia.pucpr.br
http://www.ppgia.pucpr.br/~alex
(55) (41) 330-1347

**Roberto C. S. Pacheco**
UFSC, PPGEP-CTC
C. P. 476
Florianópolis, SC.
88040-900 – Brazil
pacheco@eps.ufsc.br
(55) (48) 331-7016

## Abstract

Data mining consists of extracting knowledge from data. This paper addresses the discovery of knowledge in the form of prediction IF-THEN rules, which are a popular form of knowledge representation in data mining. In this context, we propose a new Genetic Algorithm (GA) designed specifically for discovering interesting fuzzy prediction rules. The GA searches for prediction rules that are interesting in the sense of being surprising for the user. More precisely, a prediction rule is considered interesting (or surprising) to the extent that it represents knowledge that not only was previously unknown by the user but also contradicts the original believes of the user. In addition, the use of fuzzy logic helps to improve the comprehensibility of the rules discovered by the GA, due to the use of linguistic terms that are natural for the user. The proposed GA is applied to a real-world science & technology data set, containing data about the scientific production of researchers. Experiments were performed to evaluate both the predictive accuracy and the degree of interestingness (or surprisingness) of the rules discovered by the GA, and the results were found to be satisfactory.

## 1 INTRODUCTION

The basic idea of data mining consists of extracting knowledge from data (Fayyad, 1996), (Han & Kamber, 2000). In this paper we address one general kind of data mining task, which we will refer to as the discovery of prediction rules. By prediction rule we mean an IF-THEN rule of the form:

IF <some_conditions_are_satisfied>
  THEN <predict_the_value_of_some_goal_attribute> .

Hence, we aim at discovering rules whose consequent (THEN part) predict the value of some goal attribute for an example (a record of a data set) that satisfies all the conditions in the antecedent (IF part) of the rule. We assume there is a small set of goal attributes whose value is to be predicted. The goal attributes are chosen by the user, according to his/her interest and need.

It should be noted that this task can be regarded as a generalization of the well-known classification task of data mining. In classification there is a single goal attribute to be predicted, whereas we allow more than one goal attribute to be defined by the user.

Note that, although there are several goal attributes to be predicted, each rule predicts the value of a single goal attribute in its consequent. However, different rules can predict different values of different goal attributes.

In this paper we propose a new Genetic Algorithm (GA) designed specifically for discovering interesting fuzzy prediction rules. The main motivation for using a GA in prediction-rule discovery is that GAs, due to their ability to perform a global search, tend to cope better with attribute interaction than most greedy rule induction algorithms that are traditionally used in prediction-rule discovery (Dhar et al., 2000), (Freitas, 2001).

The justification for the "interesting" and "fuzzy" characteristics of the rules is as follows. In general, fuzzy logic is a flexible way of coping with uncertainties typically found in real-world applications. In particular, in the context of data mining, fuzzy logic seems a natural way of coping with continuous (real-valued) attributes. Using fuzzy linguistic terms, such as *low* or *high*, one can more naturally represent rule conditions involving continuous attributes, by comparison with crisp discretization of those attributes. For instance, the fuzzy condition "*Salary = low*" seems more natural for a user than the crisp condition "Salary < *$14,328.53*".

Although we do use fuzzy logic to improve the comprehensibility of the rules discovered by the GA, the focus of this paper is not on the use of fuzzy logic, but rather on the discovery of "interesting" rules. We emphasize that this is a difficult problem, relatively little explored in the literature. Most algorithms for discovering prediction rules focus on evaluating the predictive accuracy of the discovered rules (Hand, 1997), without trying to discover rules that are truly interesting for the user.

It should be noted that a rule can have a high predictive accuracy but be uninteresting for the user, because it represents some obvious or previously-known piece of knowledge. A classic example is the rule:

IF <patient is pregnant> THEN <patient is female>.

Hence, a major contribution of this paper is to propose a GA that searches for rules that not only have a high predictive accuracy but also are interesting, in the sense of being surprising (representing novel knowledge) for the user. As will be seen later, the core of the GA consists of an elaborate fitness function which takes both these aspects of rule quality into account.

Another contribution of this paper is that we apply the proposed GA to the mining of a real-world science & technology data set, containing data about the scientific production of researchers (cientometric data).

The remainder of this paper is organized as follows. Section 2 reviews relevant related work. Section 3 describes in detail the proposed GA for discovering interesting (surprising) fuzzy prediction rules. Section 4 reports the results of computational experiments. Finally, section 5 concludes the paper.

## 2 RELATED WORK

### 2.1 EAs FOR DISCOVERING FUZZY PREDICTION RULES

There has been very extensive research on evolutionary algorithms (EAs) for discovering fuzzy prediction rules. Roughly speaking, the algorithms can be divided into two broad groups:

(a) EAs evolving one or more aspects of membership functions, such as the number of membership functions (linguistic terms) for each attribute, the shape of the membership functions, etc. (Xiong & Litz, 1999), (Mota et al. 1999), (Mendes et al., 2001);

(b) EAs using user-defined membership functions, and evolving only the combinations of attribute values considered relevant for predicting a goal attribute (Ishibuchi & Nakashima, 1999), (Walter & Mohan, 2000).

We follow the later approach, due to two mains reasons. First, it allows us to incorporate the domain knowledge of the user into the specification of the membership functions, leading to membership functions which are more comprehensible for the user. This is important in our data mining application, where the discovered prediction rules are directly interpreted by a human decision maker. Second, it considerably reduces the search space, since the GA has to search only for combinations of attribute values to be included in the rules.

It should be noted the above-mentioned projects focus on the discovery of fuzzy rules with high predictive accuracy, without trying to discover surprising rules. Our work differs from these projects in that the proposed GA searches for fuzzy prediction rules that are not only accurate but also surprising for the user, representing knowledge that was previously unknown by the user, as will be seen later.

### 2.2 DISCOVERING INTERESTING PREDICTION RULES

There are two broad approaches for discovering interesting rules in data mining, namely the objective approach and the subjective approach. In general, the objective approach uses a rule-discovery method and a rule-quality measure that are independent of the user and the application domain (Major & Mangano, 1993), (Noda et. al, 1999).

By contrast, the subjective approach uses a rule-discovery method and/or a rule-quality measure that take into account the background knowledge of the user about the application domain (Silberchatz & Tuzhilin, 1996), (Liu & Hsu, 1996), (Liu et al., 1997).

Hence, in general the objective approach has more generality and autonomy than the subjective approach, whereas the subjective approach has the important advantage of using the user's background knowledge to guide the search for rules. Therefore, if the application domain is well-defined and a user who is an expert in the application domain is available, it makes sense to use the subjective approach. This is the case of the project reported in this paper. The proposed GA was developed with the primary goal of mining science & technology data, a well-defined application domain, and a user expert in this application domain was available. Therefore, in this paper we follow the subjective approach.

Out of the above-mentioned projects, there are two that are more related to our research. The first one is the work of (Liu & Hsu, 1996), (Liu et al., 1997). This work follows the subjective approach. It proposes the use of general impressions to guide the search for interesting rules. General impressions can be thought of as "rules" specified by the user, representing the background knowledge and believes of the user about the application domain. (General impressions will be discussed in more detail later.) Liu and his colleagues propose the use of general impressions as the basis for a post-processing method to select the most interesting rules, among all discovered rules. That is, first a data mining algorithm is run, discovering a potentially large number of rules. Then the discovered rules are matched against the user-specified general impressions, in order to select the most interesting rules.

Our work also uses the idea of user-specified general impressions to discover interesting rules. However, it differs from the above work in that we use general impressions directly in the search for rules, rather than as a post-processing method. In other words, instead of first generating a large number of rules and then selecting the most interesting ones, the set of general impressions is directly used by the data mining algorithm to generate only interesting rules, avoiding the unnecessary generation of many rules that will be later discarded due to their lack of interestingness for the user. In addition, we propose a GA for discovering interesting rules, whereas the work of Liu and his colleagues does not use any evolutionary algorithm.

The second work related to our research is the GA for discovering interesting rules proposed by (Noda et al., 1999). This GA also searches for rules that are both accurate and interesting, according to a certain rule-interestingness measure. However, our work differs from Noda et al.'s work in two major points. First, unlike their GA, our GA discovers fuzzy rules. Second, Noda et al. follow an objective approach for the discovery of interesting rules, whereas our GA follows a subjective approach based on user-specified general impressions, as mentioned above.

# 3 A NEW GA FOR DISCOVERING INTERESTING (SURPRISING) FUZZY PREDICTION RULES

In this section we propose a new GA for discovering interesting (surprising) fuzzy rules. Hence, each individual represents a prediction rule. More precisely, each individual represents the antecedent (IF part) of a prediction rule. The consequent (THEN part) of the rule is not encoded in the genome. Rather, it is fixed for a given GA run, so that in each run all the individuals represent rules with the same consequent (value predicted for a goal attribute). Therefore, in order to discover rules predicting different goal attribute values, we need to run the GA several times, once for each value of each goal attribute.

Furthermore, the prediction rules represented by the individuals are fuzzy rules. We stress that only the rule antecedents are fuzzified. Rule consequents are always crisp. Concerning the rule antecedent, of course only conditions involving continuous (real-valued) attributes are fuzzified. Categorical (nominal) attributes are inherently crisp. For instance, there is no need to fuzzify a rule condition such as "*Sex = female*".

## 3.1 INDIVIDUAL REPRESENTATION

The genome of an individual represents a conjunction of conditions specifying a rule antecedent. Each condition is represented by a gene, and it consists of an attribute-value pair of the form $A_i = V_{ij}$, where $A_i$ is the i-th attribute and $V_{ij}$ is the j-th value belonging to the domain of $A_i$. In order to simplify the encoding of conditions in the genome, we use a positional encoding, where the i-th condition is encoded in the i-th gene. Therefore, we need to represent only the value $V_{ij}$ of the i-th condition in the genome, since the attribute of the i-th condition is implicitly determined by the position of the gene. In addition, each gene also contains a boolean flag ($B_i$) that indicates whether or not the i-th condition is present in the rule antecedent. Hence, although all individuals have the same genome length, different individuals represent rules of different lengths (which is, of course, desirable in prediction rules, since one does not know a priori how many conditions will be necessary to create a good prediction rule). The structure of the genome of an individual is illustrated in Figure 1, where m is the number of attributes of the data being mined.

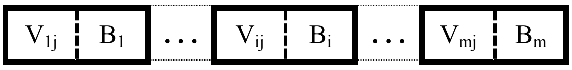

Figure 1: Genome of an individual representing a rule antecedent

We emphasize that the operator "=" is used for both fuzzy conditions and crisp conditions, as follows. As usual in the data mining and machine learning literature, our GA can cope with two kinds of attributes: continuous (real-valued) attributes and categorical (nominal) ones. Categorical attributes are inherently crisp, so that they are associated with crisp conditions such as "*Sex = female*". Continuous attributes are fuzzified, so that they are associated with fuzzy conditions such as "*Age = low*", where *low* is a fuzzy linguistic term.

## 3.2 FUZZIFYING CONTINUOUS ATTRIBUTES

Recall that, as discussed in section 2, in this work the GA uses user-defined membership functions. Hence, it evolves the combinations of attribute values considered relevant for predicting a goal attribute, but there is no need to evolve the membership functions.

In our GA the fuzzification of continuous attributes is performed as follows. Each continuous attribute is associated with either two or three linguistic terms (corresponding to the "values" of the fuzzified attribute), namely either {*low*, *high*} or {*low*, *medium*, *high*}. Each of these linguistic terms is defined by a user-specified membership function. These functions have a trapezoidal format, where there are three (or two) linguistic terms.

## 3.3 FITNESS FUNCTION

Recall that each individual is associated with a fuzzy prediction rule. In the vast majority of the literature, the main criterion used to evaluate the quality of a fuzzy prediction rule is predictive accuracy. This criterion is also important in our application, but it is not the only one. As discussed in the Introduction, a prediction rule can be accurate but not interesting for the user. This will be the case when the rule represents some relationship in the data that was already known by the user. To avoid this, our fitness function takes into account two criteria:

(a) The predictive accuracy of the rule;

(b) A measure of the degree of interestingness (or surprisingness) of the rule.

With respect to the latter criterion, our GA favors the discovery of rules that are explicitly surprising for the user, as will be seen later.

These two criteria are combined into a weighted formula as follows:

$$\text{Fitness(i)} = \text{Acc(i)} * \text{Surp(i)}$$

The measures of Acc(i) and Surp(i) are described in the next two subsections, respectively, since they are computed by separated elaborate procedures.

### 3.3.1 Measuring the Predictive Accuracy of a Fuzzy Rule

The first step to measure the predictive accuracy of a fuzzy rule is to compute the degree to which an example belongs to a rule antecedent. Recall that the rule antecedent consists of a conjunction of conditions. We use the standard fuzzy AND operator, where the degree of membership of an example to a rule antecedent is given by:

$$\min_{i=1}^{z}(\mu_i)$$

where $\mu_i$ denotes the degree to which the example belongs to the i-th condition of the rule antecedent, z is the number of conditions in the rule antecedent, and min is the minimum operator. The degree to which the example belongs to the i-th condition is directly determined by the value of the corresponding membership function for the example's attribute value associated with that condition. Of course, crisp conditions can have only either 0 or 1 membership degrees.

For instance, consider a rule antecedent with the following two rule conditions: (*Age = low*) AND (*Sex = female*), where the first condition is fuzzy and the second one is crisp. Suppose that a given example has the values *23* and *female* for the attributes *Age* and *Sex*, respectively. Suppose also that the membership function for the *low* linguistic term of *Age* returns the value 0.8 for the value *23*. Then the degree to which this example belongs to this rule antecedent is min(0.8,1.0) = 0.8.

Let A be the antecedent of a given rule. Once the degree to which each example belongs to A has been computed, the predictive accuracy of the i-th individual (fuzzy rule), denoted Acc(i), is computed by the formula:

$$Acc(i) = (CorrPred - 1/2) / (TotPred)$$

where CorrPred (number of correct predictions) is the summation of the degrees of membership in A for all examples that have the value $V_{ij}$ predicted by the rule and TotPred (total number of predictions) is the summation of the degrees of membership in A for all examples. This formula is essentially a fuzzy version of a crisp measure of predictive accuracy used by some data mining algorithms (Quinlan, 1987), (Noda et al., 1999). The rationale for subtracting 1/2 from CorrPred in the numerator is to penalize rules that are too specific, which are probably overfitted to the data. For instance, suppose CorrPred = 1 and TotPred = 1. Without subtracting 1/2 from CorrPred the modified formula would return a predictive accuracy of 100% for the rule, which intuitively is an over-optimistic estimate of predictive accuracy in this case. However, subtracting 1/2 from CorrPred the above formula returns 50%, which seems a more plausible estimate of predictive accuracy, given that the rule is too specific. Clearly, for large values of CorrPred and TotPred the subtraction of 1/2 will not have a significant influence in the value returned by the formula, so that this subtraction penalizes only rules which are very specific, covering just a few examples.

### 3.3.2 Measuring the Degree of Surprisingness of a Prediction Rule

We consider a prediction rule interesting to the extent that it is surprising for the user, in the sense of representing knowledge that not only was previously unknown but also contradicts the original believes of the user. Clearly, the problem of discovering surprising rules is a very difficult one, which has been relatively little investigated in the data mining literature. (As mentioned above, the vast majority of the literature focus on the discovery of rules with a high predictive accuracy, without trying to measure how novel or surprising the rule is for the user.)

In order to tackle this problem we follow a subjective approach for discovering surprising rules, based on the use of user-specified general impressions (Liu & Hsu, 1996), (Liu et al., 1997). In essence, a general impression specifies some relationship that the user believes to be true in the data being mined. General impressions, like prediction rules, are expressed in the form IF <conditions> THEN <predicted value>. The main difference is that general impressions are manually specified and represent believes of the user about relationships in the data, whereas prediction rules are automatically discovered and represent relationships that seem to hold in the data, according to the criteria used by the data mining algorithm. Therefore, the specification of general impressions assume that the user already has some previous knowledge or hypotheses about relationships that hold in the application domain - in our case, science and technology data.

Let $\{R_1,...,R_i,...R_{|R|}\}$ be the set of rules in the current population of the GA, where |R| denotes the number of rules (individuals); and let $\{GI_1,...,GI_j,...GI_{|GI|}\}$ be the set of general impressions representing the previous knowledge and believes of the user, where |GI| denotes the number of general impressions. Note that the set $\{GI_1,...,GI_j,...GI_{|GI|}\}$ is specified by the user before the GA starts to run, and it is kept fixed throughout the GA run. In order to compute the degrees of surprisingness of the rules in the current population, each rule is matched against every GI, as shown in Figure 2.

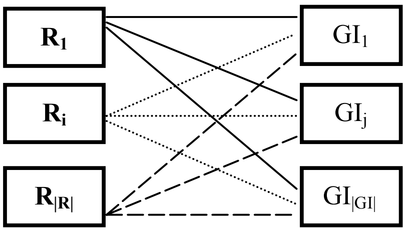

Figure 2: Matching between each rule and every general impression

A rule $R_i$ is considered surprising, in the sense of contradicting a general impression $GI_j$ of the user, to the extent that $R_i$ and $GI_j$ have similar antecedents and contradictory consequents. In other words, the larger the similarity of the antecedents of $R_i$ and $GI_j$ and the larger the degree of contradiction of the consequents of $R_i$ and $GI_j$, the larger the degree of surprisingness of rule $R_i$ with respect to general impression $GI_j$.

For each pair of rule $R_i$ and $GI_j$ - where i varies in the range 1,...,|R| and j varies in the range 1,...,|GI| - the GA computes the degree of surprisingness of $R_i$ with respect to $GI_j$ in three steps, as follows.

*First step: finding the general impressions whose consequents are contradicted by the consequent of $R_i$.* We say that the consequent of $R_i$ contradicts the consequent of a general impression $GI_j$ if and only if $R_i$ and $GI_j$ have the same goal attribute but a different goal attribute value in their consequent. For instance, this would be the case if $R_i$ predicts "*production = low*" and $GI_j$ predicts "*production = high*". Note that if $R_i$ and $GI_i$ predict different goal attributes, or if they predict the same value for the same goal attribute, there is no contradiction between them, and so the degree of surprisingness of $R_i$ with respect to $GI_i$ is considered zero, and in this case the second and third steps, described below, are ignored.

*Second step: computing the similarity between the antecedents of $R_i$ and $GI_j$.* For each general impression $GI_j$

found in the previous step (i.e, each general impression $GI_j$ contradicted by $R_i$), the system computes the similarity between the antecedents of $R_i$ and $GI_j$. This similarity, denoted $AS_{(i,j)}$, is computed by the formula:

$$AS_{(i,j)} = |A_{(i,j)}| / \max(|R_i|,|GI_j|) ,$$

where $|R_i|$ is the number of conditions (attribute-value pairs) in rule $R_i$, $|GI_j|$ is the number of conditions in general impression $GI_j$, max is the maximum operator, and $|A_{(i,j)}|$ is the number of conditions that are exactly the same (i.e., have the same attribute and the same attribute value) in both $R_i$ and $GI_j$. This formula is a somewhat simplified version of the formulas proposed by (Liu & Hsu, 1996) to measure the similarity between the antecedents of $R_i$ and $GI_j$. Those authors proposed separate formulas to measure the similarity with respect to attributes and with respect to attribute values, whereas we have chosen to incorporate both aspects of antecedent similarity into a single formula, for the sake of simplicity.

*Third step: computing the degree of surprisingness of $R_i$ with respect to $GI_j$.* Let Surp(i,j) denote the degree of surprisingness of $R_i$ with respect to $GI_j$. Surp(i,j) depends on both $AS_{(i,j)}$, computed in the second step, and on the difference between the rule consequents of $R_i$ and $GI_j$, computed in the first step, as follows. The goal attribute values in the consequents of $R_i$ and $GI_j$ can be either a value in the set {*low*, *high*} or a value {*low*, *medium*, *high*}, depending on the goal attribute. (The choice between these two attribute domains is made by the user for each goal attribute, as will be seen later.) If the difference between the consequents of $R_i$ and $GI_j$ is that one of them is *low* and the other one is *high*, characterizing the greatest possible difference between those consequents, then Surp(i,j) is assigned the value of $AS_{(i,j)}$, without any modification. If the difference between the consequents of $R_i$ and $GI_j$ is that one of them is *medium* and the other one is either *low* or *high*, characterizing a smaller difference between those consequents, then Surp(i,j) is assigned half the value of $AS_{(i,j)}$, i.e. Surp(i,j) = 0.5 x $AS_{(i,j)}$. In the latter case Surp(i,j) is assigned a smaller value than in the former case to reflect the fact that the degree of contradiction is correspondingly smaller.

Finally, once the above three steps have been completed for all general impressions, with respect to a given rule $R_i$, the system has computed all the degrees of surprisingness of $R_i$ with respect to every general impression $GI_j$, i.e. all Surp(i,j), $j$=1,...,$|GI|$, where $|GI|$ is the number of general impressions. At this point the degree of surprisingness of rule $R_i$, denoted Surp(i), is simply computed by the formula:

$$Surp(i) = \max_{j=1}^{|GI|}\left[AS_{(i,j)}\right]$$

where max returns the maximum value among its arguments.

### 3.4 SELECTION AND GENETIC OPERATORS

The GA uses tournament selection (Blickle, 2000), which essentially works as follows. First, $k$ individuals are randomly picked (k = 2), with replacement, from the population. Then the individual with the best fitness value, out of the $k$ individuals, is selected as the winner of the tournament. This process is repeated $P$ times, where $P$ is the population size. Next the $P$ selected individuals undergo genetic operators, as follows.

The GA uses relatively simple crossover and mutation operators. It uses uniform crossover (Goldberg, 1989). There is a probability for applying crossover to a pair of individuals and another probability for swapping each corresponding pair of gene (attribute)'s value in the genome of two individuals. The crossover probabilities used were 0.85 for the crossover operator and 0.5 for attribute value swapping. Our choice of uniform crossover was motivated by the fact that this operator has no positional bias, i.e., the probability of swapping each pair of attribute values is independent of the position of that attribute value in the genome. This is desirable in our data mining application, where the rule antecedent represented by the genome consists of an unordered set of conditions.

The mutation operator randomly transforms the value of an attribute into another (different) value belonging to the domain of that attribute. The mutation probability used was 0.02.

In addition to crossover and mutation operators, the GA also uses operators that insert/remove conditions to/from a rule. In essence, the condition-insertion operator switches on the flag of some condition in the genome, rendering it present in the decoded rule antecedent. Conversely, the condition-removal operator switches off the flag of some condition in the genome, which effectively removes that condition from the decoded rule antecedent. The condition-insertion and condition-removal operators perform specialization and generalization operations in the rule, respectively. Hence, they contribute for a broader exploration of the search space, facilitating the exploration of some regions of the search space that might not be so easily accessible to crossover and mutation operators.

## 4 COMPUTATIONAL RESULTS

We now report the results of computational experiments performed with the GA proposed in the previous section. In these experiments the set of general impressions was specified by the Head of Research of the State University of Maringá (Brazil). The same user also evaluated the interestingness of the rules discovered by the GA, as will be seen later. The data set used in our experiments is described in section 4.1.

The rules discovered by the GA were evaluated with respect to two criteria, namely:

*(a) Predictive accuracy.* As usual in the literature, predictive accuracy was measured in an objective way, by computing the prediction accuracy rate on a test set separate from the training set. The results with respect to predictive accuracy are reported in section 4.2.

(b) *Degree of interestingness (surprisingness)*. This is a measure of how surprising, novel the rule is for the user, as explained in the previous section. This was measured in a subjective way, by showing the discovered rules to the user and ask him to assess them according to how interesting they were. The results with respect to interestingness are reported in section 4.3.

## 4.1 THE DATA SET

The application domain addressed in this paper involves a science and technology database obtained from CNPq (the Brazilian government's National Council of Scientific and Technological Development). More precisely, we have mined a subset of the database containing data about the scientific production of researchers of the south region of Brazil. However, it should be noted that the design of the GA is generic enough to allow its use in virtually any other application domain, as long as proper general impressions and membership functions are specified by the user.

The experiments reported in this paper have been performed with 24 attributes. The selection and preparation of these attributes for data mining purposes was a time-consuming process, taking several months, since the original data set was not collected for data mining purposes.

The data set contained 5,690 records (examples), and each record had attributes describing a given researcher and his scientific production in the period from 1997 to 1999. Records that had any attribute with missing value were removed. Out of the 24 attributes, 6 were used as goal attributes to be predicted, and the other 18 attributes were used as predictor attributes. Out of the 18 predictor attributes, 8 were categorical (nationality, continent of origin, sex, state, city, skill in writing English, whether or not she/he was the head of a research group, main research area) and 10 were continuous (educational level, No. of years since last graduation, age, No. of completed technical projects, No. of delivered courses, No. of supervised Ph.D. thesis, No. of supervised M.Sc. dissertations, No. of supervised research essays (at the diploma level), No. of supervised final-year undergraduate projects, No. of supervised undergraduate students with a research scholarship). The 10 continuous attributes were fuzzified for rule-discovery purposes, as previously explained.

For prediction purposes, each goal attribute was discretized into either two values (referring to a low or high scientific production) or three values (referring to a low, medium or high scientific production), as determined by the user.

The 6 goal attributes, denoted $G_1,...,G_6$, have the following meaning and values to be predicted:

*$G_1$ = No. of papers published in national journals - values: low, medium, high;*
*$G_2$ = No. of papers published in internat. journals - values: low, medium, high;*
*$G_3$ = No. of chapters published in national books - values: low, medium, high;*
*$G_4$ = No. of chapters published in international books - values: low, high;*
*$G_5$ = No. of national edited/published books - values: low, high;*
*$G_6$ = No. of internat. edited/published books - values: low, high.*

Therefore, in total there are 15 goal attribute values to be predicted.

## 4.2 EVALUATING THE PREDICTIVE ACCURACY OF DISCOVERED RULES

In order to measure the predictive accuracy of discovered rules, we have performed a well-known 10-fold cross-validation procedure (Hand, 1997). In essence, this procedure works as follows. First, the data set is divided into 10 mutually exclusive and exhaustive partitions. Then the data mining algorithm is run 10 times. In the i-th run, i=1,...,10, the i-th partition is used as the test set, and the remaining 9 partitions are temporarily grouped and used as the training set. In each run the system computes the prediction accuracy rate on the test set, which is the ratio of the number of correct predictions over the total number of predictions. The reported result is the average prediction accuracy rate over the 10 runs.

We have compared the predictive accuracy of the rules discovered by our GA with the predictive accuracy of the rules discovered by J4.8 (Witten, 2000). The latter is a decision-tree-building algorithm which is included in a public-domain data mining tool available at: www.cs.waikato.ac.nz/ml/weka/index.html. J4.8 is a modified version of the very well-known decision-tree-building algorithm C4.5 (Quinlan, 1993).

Note that J4.8 (as well as C4.5) is an algorithm designed for the classification task of data mining, where there is a single goal attribute to be predicted. Similarly, each run of our GA discovers a rule predicting a different goal attribute value. Hence, both J4.8 and our GA have to be run several times in our application, since we are interested in discovering rules predicting several goal attributes. More precisely, J4.8 was "run" 6 times (each "run" actually consists of the 10 runs of a 10-fold cross-validation procedure), whereas our GA was "run" 15 times (again, each "run" was a 10-fold cross-validation procedure), corresponding to the 15 different goal attribute values for all the 6 goal attributes.

Note also that J4.8 and our GA were designed for discovering different kinds of prediction rules. The two main differences are as follows. First, J4.8 just tries to discover accurate rules. It does not try to discover interesting, surprising rules. By contrast, our GA tries to discover rules that are both accurate and surprising for the user. Second, J4.8 was designed for discovering classification rules covering all examples. That is, given any test example, J4.8 must have discovered a rule that can be used to predict its class. By contrast, our GA does not try to discover rules covering all examples. It tries to discover only a small set of interesting, surprising rules, the knowledge "nuggets". The discovered rules can collectively cover only a relatively small subset of examples, and yet be considered surprising, high-quality rules. These two differences make it difficult to compare the two algorithms in a fair way.

In order to make this comparison more fair, we have eliminated the above first difference. This was achieved by modifying the fitness function of the GA (only in the experiments reported in this section) so that the fitness of an individual (rule) is measured only by its predictive accuracy, ignoring its degree of surprisingness, i.e.:

$$\text{Fitness}(i) = \text{Acc}(i) = (\text{CorrPred} - 1/2) / (\text{TotPred})$$

Now both J4.8 and the GA search only for accurate rules.

The above second difference between the two algorithms is more difficult to eliminate, and it still remains a difference in our experiments. This problem will be the subject of future research.

The predictive accuracy obtained by J4.8 and our GA is reported in Table 1. The first column of this table identifies the goal attribute predicted by the rule (see the meaning of $G_1$...$G_6$ in the previous section), whereas the second column identifies the value predicted for that goal attribute. The third column identifies the relative frequency (in %) of the corresponding goal attribute value in the training set. The fourth and fifth columns report the prediction accuracy rate (in %) in the test set (10-fold cross-validation) of J4.8 and the GA, respectively. In each row, we show in bold the larger predictive accuracy rate, out of the rates obtained by the two algorithms.

Table 1: Prediction Accuracy Rate (%) of J4.8 and GA

| **Goal attrib.** | **Predicted value** | **Freq. (%)** | **J4.8** | **GA** |
|---|---|---|---|---|
| $G_1$ | low | 46.9 | **64.9** | 58.8 |
| | medium | 50.6 | **63.9** | 60.4 |
| | high | 2.5 | **9.1** | 0.0 |
| $G_2$ | low | 64.2 | 76.6 | **90.7** |
| | medium | 29.7 | **45.3** | 40.0 |
| | high | 6.1 | **32.2** | 25.0 |
| $G_3$ | low | 76.9 | 82.2 | **95.2** |
| | medium | 21.2 | 45.3 | **56.7** |
| | high | 1.9 | **27.4** | 25.0 |
| $G_4$ | low | 93.2 | 93.4 | **98.4** |
| | high | 6.8 | **51.7** | 14.3 |
| $G_5$ | low | 83.5 | 86.0 | **89.5** |
| | high | 16.5 | 54.7 | **56.9** |
| $G_6$ | low | 97.9 | 97.9 | **98.9** |
| | high | 2.1 | 0.0 | 0.0 |

As can be seen in the table, the prediction accuracy rate of the GA is larger than the one of J4.8 in seven rows (i.e., seven goal attribute values), whereas the converse is true in other seven rows. With the exception of the goal attribute $G_1$, in general the GA outperformed J4.8 in the prediction of goal attribute values with a larger frequency in the training set, whereas J4.8 outperformed the GA in values with a smaller frequency in the training set.

In any case, the focus of our experiments is the evaluation of the degree of interestingness of the rules discovered by the GA, reported in the next section.

### 4.3 EVALUATING THE INTERESTINGNESS OF THE RULES DISCOVERED BY THE GA

The rules discovered by the GA were also evaluated with respect to their degree of interestingness (surprisingness) for the user. In this experiment it was not possible to compare the GA with J4.8, since J4.8 was not designed to discover interesting rules. Actually, for the majority of the 6 goal attributes, J4.8 produced a very large decision tree, with literally hundreds of nodes. Therefore, it was not even feasible to show all rules discovered by J4.8 to the user, anyway. By contrast, the GA was explicitly designed to discover a small set of interesting rules (one rule per goal attribute value to be predicted), so that it was very feasible to show all rules discovered by the GA to the user, for his subjective evaluation.

We emphasize that the user who evaluated the interestingness of the discovered rules was the same user who specified the general impressions, as mentioned above. Actually, when the user was shown a discovered rule, he was also shown his own general impression contradicted by that rule.

The user was asked to assign to each rule discovered by the GA one of the following three degrees of interestingness (surprisingness): low interestingness, medium interestingness or high interestingness. The results of the evaluation performed by the user is reported in Table 2. The rule consequent in the first column consists of an attribute-value pair "$G_i$= *val*" identifying the goal attribute value predicted by the rule, where $G_i$ denotes the *i*-th attribute, *i*=1,...,6 (see section 4.1 for the meaning of these goal attributes) and *val* denotes the value predicted for the corresponding goal attribute. The second column of this table shows the degree of interestingness assigned to the rule by the user.

Table 2: Interestingness of rules discovered by the GA

| **Rule consequent** | **interestingness for the user** |
|---|---|
| $G_1$ = low | high |
| $G_1$ = medium | medium |
| $G_1$ = high | medium |
| $G_2$ = low | high |
| $G_2$ = medium | medium |
| $G_2$ = high | low |
| $G_3$ = low | high |
| $G_3$ = medium | low |
| $G_3$ = high | low |
| $G_4$ = low | medium |
| $G_4$ = high | medium |
| $G_5$ = low | high |
| $G_5$ = high | low |
| $G_6$ = low | high |
| $G_6$ = high | low |

The experiment reported in this section, involving 15 runs of the GA (one for each goal attribute value being predicted) took about 6 minutes. Each run of the GA had a population size of 100 individuals, which evolved during 60 generations.

The results reported in Table 2 were obtained by using the entire data set (i.e., all the 5,690 examples) as input data for the GA. This procedure is justified because when measuring the degree of interestingness of discovered rules there is no need for dividing the data into training and test sets, since there is no need for measuring predictive accuracy in the test set (which was already measured in the experiments reported in the previous section).

Out of the 15 rules discovered by the GA, 5 were assigned a high degree of interestingness by the user, 5 were assigned a medium degree of interestingness, and the remaining 5 were assigned a low degree of interestingness. Overall, this seems to be a relatively good result, considering how difficult it is to discover very interesting, surprising rules.

We have observed that there is a relationship between a rule's simplicity (in the sense of having a small number of conditions) and its degree of interestingness for the user. This relationship is due to an interaction between the measure of rule surprisingness used in this work and the kind of general impressions specified by the user, as follows. In our experiments, the user specified mainly short general impressions, having a small number of conditions. As a result, the measure of rule surprisingness favors the discovery of short rules too, since these rules can have a larger degree of similarity between the rule antecedent and the general impression antecedent.

## 5 CONCLUSIONS AND FUTURE WORK

We have proposed a GA for discovering interesting fuzzy prediction rules. The proposed GA was evaluated with respect to both the predictive accuracy and the interestingness of the discovered rules. With respect to the former criterion, the performance of the GA was compared with J4.8, a well-known decision-tree-building algorithm. Overall, the GA was found to be competitive with J4.8 with respect to this criterion.

In any case, the main focus of our experiments was on the discovery of rules that are interesting, in the sense of representing surprising, previously-unknown knowledge for the user. In our experiments the application domain was science & technology data, and the user was an expert in this domain. Overall, the GA was able to found several rules that were considered very interesting by the user. For instance, one of the general impressions specified by the user represented his previous knowledge (or belief) that biology researchers of a given region had a high number of international edited/published books. However, the GA was able to found an accurate rule contradicting this general impression. The rule had the same antecedent as the general impression but made the opposite prediction, i.e. it predicted that the researchers in question had a low number of international edited/published books. This rule was considered very interesting by the user.

The main direction for future research will be to compare the degree of interestingness of the rules discovered by our GA with the degree of interestingness of the rules discovered by another data mining algorithm that was specifically designed for the discovery of interesting rules.

### References

T. Blickle (2000). Tournament selection. In: T. Back, D. B. Fogel and Z. Michalewicz (Eds.) *Evolutionary Computation 1: Basic Algorithms and Operators*. Chapter 24. Institute of Phisics Publishing.

V. Dhar, D. Chou and F. Provost (2000). Discovering Interesting Patterns for Investment Decision Making with GLOWER – A Genetic Learner Overlaid With Entropy Reduction. *Data Mining and Knowledge Discovery 4(4)*, 251-280.

U. M. Fayyad, G. Piatetsky-Shapiro and P. Smyth (1996). *Advances in knowledge discovery & data mining*. Chapter 1: From data mining to knowledge discovery: an overview. AAAI/MIT.

A. A. Freitas (2001). Understanding the Crucial Role of Attribute Interaction in Data Mining. *Artificial Intelligence Review 16(3)*, Nov. 2001, pp. 177-199.

D. E. Goldberg (1989). *Genetic algorithms in search, optimization, and machine learning*. New York: Addison-Wesley Publishing Company, Inc.

J. Han and M. Kamber (2000). *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers.

D. J. Hand (1997). *Construction and Assessment of Classification Rules*. John Wiley & Sons.

H. Ishibuchi and T. Nakashima (1999). Designing Compact Fuzzy Rule-Based Systems with Default Hierarchies for Linguistic Approximation. *CEC-99*, p. 2341-2348.

B. Liu and W. Hsu (1996). Post-analysis of learned rules. *AAAI-96*, p. 828-834.

B. Liu, W. Hsu and S. Chen (1997). Using general impressions to analyze discovered classification rules. *Third Int. Conf. on Knowledge Discovery and Data Mining, KDD-97*, p. 31-36.

J. A. Major and J. J. Mangano (1993). Selecting among rules induced from a hurricane database. *Knowledge Discovery in Databases Workshop at AAAI-93*, p. 28-44.

R. R. F. Mendes, F. B. Voznika, A. A. Freitas and J. C. Nievola. (2001) Discovering fuzzy classification rules with genetic programming and co-evolution. Principles of Data Mining and Knowledge Discovery (Proc. 5th European Conf., PKDD 2001) - Lecture Notes in Artificial Intelligence 2168, pp. 314-325. Springer-Verlag.

C. Mota, H. Ferreira and A. Rosa (1999). Independent and Simultaneous Evolution of Fuzzy Sleep Classifiers by Genetic Algorithms. *GECCO-99*, p. 1622-1629.

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

J. R. Quinlan (1987). Generating production rules from decision trees. *Proc. IJCAI-87*, p. 304-307.

J. R. Quinlan (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

A. Silberschatz and A. Tuzhilin (1996). What Makes Patterns Interesting in Knowledge Discovery Systems. *IEEE Transactions on Knowledge and data engineering*, Vol. 8, No. 6, pp. 970-974.

D. Walter and C. K. Mohan. (2000) ClaDia: a fuzzy classifier system for disease diagnosis. *Proc. Congress on Evolutionary Computation (CEC-2000)*. La Jolla, CA, USA.

I. H. Witten and E. Frank (2000). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann Publishers.

N. Xiong and L. Litz (1999). Generating Linguistic Fuzzy Rules for Pattern Classification with Genetic Algorithms. *PKDD-99*, p. 574-579.