CONSTRAINED-SYNTAX GENETIC PROGRAMMING FOR KNOWLEDGE DISCOVERY IN MEDICAL DATABASES

Celia C. Bojarczuk [1], Heitor S. Lopes [2], Alex A. Freitas [3]
[1] Electrotechnics Department & Bioinformatics Laboratory – CEFET-PR
[2] Electronics Department & Bioinformatics Laboratory – CEFET-PR
[3] Graduate Program in Computer Science – PUC-PR
Curitiba (PR) – BRAZIL
{celia, hslopes}@cpgei.cefetpr.br, alex@ppgia.pucpr.br

Abstract
This work is intended to discover classification rules for diagnosing certain pathologies. In order to discover these rules it was used genetic programming as well as some concepts of data mining, particularly the emphasis on the discovery of comprehensible knowledge.

1. Introduction

There is a growing interest in the area of data mining, where the objective is discovering knowledge not only correct, but also comprehensible and interesting for users (Fayyad et al, 1996). In such a way, knowledge contained in databases can be extracted and combined with a user's own knowledge in order to better support the decision-making process.
In a database it is possible that there are several attributes related to each other and the user is not aware of this. In this case, data mining techniques can be employed to discover high-level rules that identify unknown relationship between attributes. The use of Genetic Programming - GP (Koza, 1992) to discover comprehensible classification rules, in the spirit of data mining, is sparsely found in the recent literature. We believe that this is a very promising area of research, since GP has demonstrated to be efficient in problems with large search spaces. In the particular case of data mining, GP can be used to search the space of all possible values of attributes and its combinations that make up classification rules for a given database.
This paper is organized as follows. Section 2 presents a brief overview of genetic programming. Section 3 describes our genetic programming system for discovering classification rules. Section 4 reports computational results. Section 5 concludes the paper.

2. A brief overview of genetic programming

Genetic programming is a powerful search method inspired by natural selection. The basic idea is to evolve a population of ``programs" candidate to the solution of a specific problem. A program (an individual of the population) is usually represented in the form of a tree, where the internal nodes are functions (operators) and the leaf nodes are terminal symbols. Both the function set and the terminal set must contain symbols appropriate for the target problem. For instance, the function set can

contain arithmetic operators, logic operators, mathematical functions, etc; whereas the terminal set can contain the variables (attributes) of the target problem.
Each individual of the population is evaluated with respect to its ability to solve the target problem. This evaluation is performed by a fitness function. Then the individual undergoes the action of genetic operators such as reproduction and crossover. The reproduction operator selects individuals of the current population in proportion to their fitness values, so that the fitter an individual is the higher the probability that it will take part in the next generation of individuals. The crossover operator replaces a randomly selected subtree of an individual with a randomly chosen subtree from another individual.
Once reproduction and crossover have been applied according to given probabilities, the newly created generation of individuals is evaluated by the fitness function. This process is repeated iteratively, usually for a fixed number of generations. The result of genetic programming (the best solution found) is the fittest individual produced along all generations.

3. A genetic programming system for discovering classification rules

This work addresses the task of classification in the context of data mining. In this context, knowledge is usually represented in the form of 'IF-THEN' rules. The 'IF' part of the rule is the antecedent and contains a combination of attributes in disjunctive normal form. The 'THEN' part is the consequent and contains the class that will be predicted for a database record that satisfies the rule antecedent. For instance, in a medical domain, considering a database of medical records, the antecedent could be a combination of symptoms and the consequent a given disease.
In the case of this work, individuals represent the antecedent of a rule, but not the consequent, since for each GP run the consequent is fixed for all individuals. Both the function set and the terminal set must contain symbols appropriate for the target problem. The function set in this work contains the logical and relational operators shown in table 1 and the terminal set contains the attributes of the database.
Due to the closure property of GP, every function should accept as arguments any combination of attributes or values returned from other functions. This property imposes serious limitations to GP when mining rules in a database, since databases have attributes of different data types. To circumvent this problem, a constrained-syntax GP was developed. Thus, some restrictions should be considered for the antecedent, in order to have a valid rule. First, a child node of an 'AND' function cannot be either an 'OR' function nor an attribute or attribute value. In other words, a child node of an AND function must be either an AND function or a comparison operator in $\{\geq, <, =, \neq\}$. Second, a child of comparison operator must be either an attribute or an attribute value. Third restriction is concerned to the uniqueness of an attribute in a rule antecedent, i.e. each attribute can occur only once in a given rule antecedent. This is implemented to avoid inconsistent rules such as: "IF (Sex=male) AND (Sex=female) ....". The last restriction is related to the input and output of the logical and relational functions (operators), which are shown in Table 1. This table specifies, for each operator (function), what are the valid data types for its input arguments and its output.

3.1. Fitness function

The fitness function evaluates the quality of an individual (rule). This means that a number of records of the database are classified by the rule under evaluation and its predictive performance is measured. The fitness function used in this paper follows the work of Bojarczuk, Lopes and Freitas (2000).

Table 1: Input and output arguments of operators used.

Operators (functions)	**Data type of input arguments**	**Data type of output**
$\geq, <$	(real, real)	boolean
$=, \neq$	(categorical, categorical)	boolean
AND, OR, NOT	(boolean, boolean)	boolean

Before we can define the fitness function, it is necessary to recall a few basic concepts on classification-rule evaluation. When using a rule for classifying an example, depending on the class predicted by a rule and on the true class of the patient (database record), four types of results can be observed for the prediction, as follows:

- true positive (*tp*)- the rule predicts that the patient has a given disease and the patient does have that disease;
- false positive (*fp*) - the rule predicts that the patient has a given disease but the patient does not have it;
- true negative (*tn*)- the rule predicts that the patient does not have a given disease, and indeed the patient does not have it;
- false negative (*fn*)- the rule predicts that the patient does not have a given disease but the patient does have it.

The fitness function used in this work combines two indicators that are commonplace in the medical domain, namely the sensitivity (*Se*) and the specificity (*Sp*), defined as follows:

$$Se = tp / (tp + fn) \quad (1)$$

$$Sp = tn / (tn + fp) \quad (2)$$

where *tp, fp, tn and fn* are variables whose value is the number of patients observed in each corresponding kind of prediction result, as defined above.

In practice, conventional GP does not produce simple solutions. Considering that the comprehensibility of a rule is inversely proportional to its size, something has to be done to enforce GP to produce rules as short as possible. Hence, we define a measure of simplicity (*Sy*) of a rule, given in equation 3:

$$Sy = \frac{(maxnodes - 0.5 * numnodes - 0.5)}{(maxnodes - 1)} \quad (3)$$

Where *numnodes* is the current number of nodes (functions and terminals) of an individual, and *maxnodes* is a parameter of the GP regarding the maximum allowed size of the tree. Equation 3 produces its maximum value of 1 when a rule is so simple that it contains just one term. This equation produces its minimum value of 0.5 when the number of nodes equals the maximum allowed. The reason to set the lower limit to 0.5 is to penalize large-sized individuals without forcing them to disappear. This is especially important in the early generations of a run, when most individuals will

have very low predictive accuracy, but can carry good genetic material capable of being improved by genetic operators.
Finally, the fitness function used by our GP is defined as the product of the indicators of predictive accuracy and simplicity, i.e.:

$$fitness = Se * Sp * Sy \quad (4)$$

Therefore, the goal of our GP is to maximize both the *Se* and the *Sp*, and minimize the rule size at the same time. This is an important point, since it would be relatively trivial to maximize the value of one of these indicators at the expense of significantly reducing the values of the others. Furthermore, the above fitness function has the advantages of being simple and returning a meaningful, normalized value in the range [0..1]. For further analysis of the motivation for maximizing the product *Se*Sp*, regardless of rule simplicity and independently of any evolutionary algorithm, see Hand (1997).

4. Computational results

Three experiments were done using databases of medical domains: chest pain, dermatology and breast cancer. The last two can be found in the Internet (Blake & Merz, 1998) and the first was used in Lopes (1999).
The chest pain database has 12 classes, corresponding to diseases commonly related to chest pain, and has 138 records with 161 attributes (binary and categorical) each. Attributes are related to the characteristics of chest pain, including symptoms, signals, clinical history and laboratory tests.
The dermatology database (Demiroz et al, 1998) has 6 classes related to dermatological pathologies, and contains 366 records, with 34 attributes each. All attributes have values mapped in the range [0..3], except age (integer, in years) and family history (0,1).
The breast cancer database is extensively used by machine learning researchers and is related to the recurrence of breast cancer of patients that undergo surgery. The database has 286 records, with 9 attributes each, and only two classes.
In all three experiments, the database was divided into five partitions, being 1/5 for testing and 4/5 for training Then a well-known 5-fold cross-validation procedure was performed. For each experiment (data set), GP was run once for each class. Once all runs of GP for a given data set were completed, all the rules found by GP in that experiment were grouped into a rule set, i.e. the set of all discovered rules (for all classes) for that data set. The quality of that rule set was evaluated according to two criteria: the predictive accuracy of the rule set and the comprehensibility of the rules.
The predictive accuracy of a rule set was measured by its classification accuracy on the test set, as usual. This seems to be the most widely used measure of predictive accuracy in the literature, in spite of its drawbacks (Hand, 1997). The results obtained for the three databases are shown in Table 2. The second column shows the results for the constrained-syntax genetic programming (GP) system proposed in this paper. The numbers after the "±" symbol are the standard deviations of the corresponding accuracy rates. The third column shows the results for the genetic algorithm (GA) proposed by (Fidelis et al. 2000). In their paper the authors report results for the Dermatology and Breast Cancer data sets, but not for the Chest Pain data set. The proposed GP obtained accuracy rates somewhat better than the GA in both the Dermatology and the Breast Cancer data sets.

Table 2: Classification accuracy rate (%) for the three medical databases.

Database	GP of this paper	GA of (Fidelis et al. 2000)
Chest pain	80,31 ± 7,80	N/A

Dermatology	96,64 ± 2,27	94.96
Breast cancer	71,79 ± 9,36	67.39

In this work, we are interested not only in the predictive accuracy of the discovered rules, but also in the comprehensibility of the rules - in the spirit of data mining. Rule comprehensibility is significantly more difficult to measure in an objective way, in comparison with predictive accuracy. There is a considerable subjective aspect in the former. In most of the literature, however, rule comprehensibility is associated with syntactic complexity. In general, the smaller the number of rules and the shorter (the smaller the number of conditions of) the rules the better. In this paper we also follow this principle for evaluating the comprehensibility of the discovered rules.

Concerning the number of discovered rules, our GP system (by design) provides the best possible result, which is exactly one rule for each class. This minimization of the number of discovered rules saves the potentially precious time of the human user, who is required to analyze only the very best rule found for each class. In contrast, some data mining algorithms may easily overload the user with a large number of discovered rules, which can hardly be considered "comprehensible" knowledge. As a striking evidence of the simplicity of the rules, they are shown in table 3, 4 and 5. These rules are considerably smaller than the rules discovered by the GA proposed by (Fidelis et al. 2000), which also discovers a single rule for each class.

Table 3: Results of learning from the full chest pain data set

Class	Discovered rule	Class	Discovered rule
1	(eco_acs or irrad_mse)	7	Rx_aep
2	Me_nitrato	8	eco_dp
3	Ecg_essst	9	(da_ulcera or fatmel_ali)
4	Rx_daad	10	azia
5	Ecg_bvc	11	(fatdes_com or dor_musc)
6	Fr_imob	12	Me_ansiol

Table 4: Results of learning from the full breast cancer data set

Class	Discovered rule
1	(deg_malig ≠ 2)
2	((inv_nodes ≥1) or (tumor_size = 5))

Table 5: Results of learning from the full dermatology data set

Class	Discovered rule	Class	Discovered rule
1	(clubbing ≠ 0)	4	(spongiosis ≠ 0)
2	(spongiosis ≥ 2)	5	(fibrosis ≠ 0)
3	(band_like ≥ 2)	6	(perifollicular ≠ 0)

5. Conclusions

In this paper we have proposed a constrained-syntax genetic programming system for discovering high-level, comprehensible classification rules. Results obtained so far are quite promising. From a

predictive accuracy viewpoint, the results are similar to results found in the literature using other approaches. A remarkable result obtained by proposed genetic programming system is the comprehensibility of the discovered rule set: essentially one rule per class and with a very small quantity of attributes (conditions) in each rule.

References

BLAKE C.L., MERZ C.J., (1998) UCI Repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science, [http://www.ics.uci.edu/~mlearn/MLRepository.html].

BOJARCZUK C.C., LOPES H.S., FREITAS A.A., (2000) Genetic programming for knowledge discovery in chest-pain diagnosis. *IEEE Engineering in Medicine and Biology Magazine*, **19**, 4, 38-44.

DEMIROZ G., GOVENIR H.A., ILTER N., (1998) Learning differential diagnosis of eryhemato-squamous diseases using voting feature intervals. *Artificial Intelligence in Medicine*, **13**, 147-165.

FAYYAD U.M., PIATETSKY-SHAPIRO G., SMYTH P., (1996) From data mining to knowledge discovery: an overview. In: Fayyad, U.M. et al (eds.), *Advances in Knowledge Discovery and Data Mining*. Cambridge: MIT Press, 1-33.

FIDELIS M.V., LOPES, H.S., FREITAS A.A., (2000) Discovering comprehensible classification rules with a genetic algorithm. *Proc. 2000 Congress on Evolutionary Computation (CEC-2000)*, 805-810. La Jolla, CA, USA.

HAND D., (1997) *Construction and Assessment of Classification Rules*. New-York: John Wiley, & Sons.

KOZA J.R., (1992) *Genetic Programming: on the Programming of Computers by Means of Natural Selection*. Cambridge: MIT Press.

LOPES H.S., (1999) A Medical Diagnostic System Optimization Using Parallel Genetic Algorithms. In: Szczepaniak P.S. (ed.), *Computational Intelligence and Applications*. Heidelberg, Physica-Verlag, 222-227.