Data Mining with Constrained-Syntax Genetic Programming: Applications in Medical Data Set

Celia C. Bojarczuk[a], Heitor S. Lopes[b], Alex A. Freitas[c]

[a] *Electrotechnics Department & Bioinformatics Laboratory – CEFET-PR, Curitiba, PR, Brazil*
[b] *Electronics Department & Bioinformatics Laboratory – CEFET-PR, Curitiba, PR, Brazil*
[c] *Graduate Program in Computer Science – PUC-PR, Curitiba, PR, Brazil*

Abstract

This work is intended to discover classification rules for diagnosing certain pathologies. In order to discover these rules we have developed a new constrained-syntax genetic programming algorithm based on some concepts of data mining, particularly with emphasis on the discovery of comprehensible knowledge. We compare the performance of the proposed GP algorithm with a genetic algorithm and with the very well-known decision-tree algorithm C4.5.

Keywords:

Genetic Programming, Data Mining, Classification, Constrained-Syntax Genetic Programming

Introduction

In essence, data mining consists of discovering knowledge that is not only correct but also comprehensible and interesting for users [4]. Therefore, discovered knowledge can be combined with a user's own knowledge in order to better support the decision-making process. In this paper we use Genetic Programming - GP [9] – to discover comprehensible classification rules from data. 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 addition, GP performs a global search in the space of candidate rules, which intuitively makes it to cope better with attribute interaction than local search algorithms [6][7].

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.

A Brief Overview of Genetic Programming

Genetic programming is a powerful search method inspired by natural selection [9]. 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.

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/or signals 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. 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.

Table 1: Operators and the data types of their arguments

Operators (functions)	Data type of input arguments	Data type of output
≥, <	(real, real)	Boolean
=, ≠	(categorical, categorical)	Boolean
AND, OR, NOT	(boolean, boolean)	Boolean

Due to the closure property of GP [9], every function should accept as arguments any combination of attributes or values returned from other functions. This property imposes serious limitations to GP when discovering 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 {≥, <, =, ≠}. Second, a child node of a comparison operator must be either an attribute or an attribute value. The third restriction is concerned with 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.

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 [2]. The fitness function can be thought of as consisting of two parts. First, it 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 the number of true positives, false positives, true negatives and false negatives, respectively – see [8]. The second part of the fitness function is a term that enforce GP to produce rules as short as possible. The motivation is that the comprehensibility of a rule is inversely proportional to its size. Hence, we define a measure of simplicity (*Sy*) of a rule as follows.

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

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. This equation 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 later 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.:

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

Computational results

Three experiments were done using databases of medical domains: chest pain, dermatology and Ljubljana breast cancer. The two latter can be found in the Internet [1] and the first was used in [10][2]. 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 [3] has 6 classes related to dermatological pathologies, 366 records and 34 attributes. 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 related to the recurrence of breast cancer of patients that undergo surgery. It has 286 records, 9 attributes, and two classes.

In all three experiments, the database was randomly divided into five partitions. Then a well-known 5-fold cross-validation procedure was performed. In each iteration of the cross-validation procedure one of the 5 partitions was used as the test set and the other 4 partitions were used as the training set. The predictive accuracy results reported below are an average over all five iterations of the cross-validation procedure.

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 [8]. 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 [5]. 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.

The fourth column of Table 2 shows the results for C4.5 [11], a very well-known, advanced decision-tree algorithm often used in machine learning and data mining. The results for C4.5 were also obtained by a 5-fold cross-validation procedure. The proposed GP obtained accuracy rates considerably better than C4.5 in the Chest Pain and Dermatology data sets. The two algorithms obtained similar accuracy rates in the Breast Cancer data set. Overall, the rule set discovered by the GP was simpler (shorter) than the rule sets discovered by both the above-mentioned GA and C4.5.

Table 2: Classification accuracy rate (%) for three medical data sets.

Data set	Proposed GP	GA	C4.5
Chest pain	80,31 ± 7,80	N/A	73.18
Dermatology	96,64 ± 2,27	94.96	89.12
Breast cancer	71,79 ± 9,36	67.39	71.38

Conclusions

In this paper we have proposed a constrained-syntax genetic programming (GP) system for discovering high-level, comprehensible classification rules. The proposed GP was compared with a genetic algorithm (GA) and with C4.5. Results obtained so far are very promising. Overall the GP discovered rule sets that are more accurate and simpler than the rule sets discovered by both the GA and C4.5. In the future we intend to apply the proposed GP to other data sets, to further validate the results reported in this paper.

References

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

[2] Bojarczuk C.C., Lopes H.S., Freitas A.A., Genetic programming for knowledge discovery in chest-pain diagnosis. *IEEE Engineering in Medicine and Biology Magazin*e, 2000: **19**, v. 4, pp. 38-44.

[3] Demiroz G., Govenir H.A., Ilter N., Learning differential diagnosis of eryhemato-squamous diseases using voting feature intervals. *Artificial Intelligence in Medicin*e, 1998: **13**, pp. 147-165.

[4] Fayyad U.M., Piatetsky-Shapiro G., Smyth P., From data mining to knowledge discovery: an overview. In: Fayyad, U.M. et al (eds.), *Advances in Knowledge Discovery and Data Minin*g. Cambridge: MIT Press, 1996, pp. 1-33.

[5] Fidelis M.V., Lopes, H.S., Freitas A.A., Discovering comprehensible classification rules with a genetic algorithm. *Proc. 2000 Congress on Evolutionary Computation (CEC-2000)*, 2000, pp. 805-810. La Jolla, CA, USA.

[6] Freitas A.A. Evolutionary Algorithms. To appear in: Zytkow, J. and Klosgen, W. (Eds.) *Handbook of Data Mining and Knowledge Discovery.* Oxford University Press, 2001.

[7] Freitas A.A. A Survey of Evolutionary Algorithms for Data Mining and Knowledge Discovery. To appear in: Ghosh, A. and Tsutsui, S. (Eds.) *Advances in Evolutionary Computation.* Springer-Verlag, 2001.

[8] Hand D., *Construction and Assessment of Classification Rule*s. New-York: John Wiley, & Sons, 1997.

[9] Koza J.R., *Genetic Programming: on the Programming of Computers by Means of Natural Selection.* Cambridge: MIT Press, 1992.

[10]Lopes H. S., A Medical Diagnostic System Optimization Using Parallel Genetic Algorithms. In: Szczepaniak P.S. (ed.), *Computational Intelligence and Applications*. Heidelberg, Physica-Verlag, 2000, pp. 222-227.

[11]Quinlan, J.R. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

Address for correspondence

Célia Cristina Bojarczuk
Centro Federal de Educação Tecnológica do Paraná - CEFET-PR
Departamento Acadêmico de Eletrotécnica - DAELT
Av. 7 de setembro, 3165, Bloco D, 1º andar
80230-901 - Curitiba - Paraná - Brasil
E-mail: celia@cpgei.cefetpr.br

Heitor Silvério Lopes
Centro Federal de Educação Tecnológica do Paraná - CEFET-PR
Programa de Pós-Graduação em Engenharia Elétrica e
Informática Industrial - CPGEI
Av. 7 de setembro, 3165, Bloco D, 3º andar
80230-901 - Curitiba - Paraná - Brasil
E-mail: hslopes@cpgei.cefetpr.br
URL adress: http://www.cpgei.cefetpr.br/~hslopes

Alex A. Freitas
Pontificia Universidade Catolica do Paraná - PUCPR
Programa de Pos-Graduacao em Informatica Aplicada - PPGIA
Rua Imaculada Conceicao, 1155
80215-901 Curitiba - PR - Brasil
E-mail: alex@ppgia.pucpr.br
URL adress: http://www.ppgia.pucpr.br/~alex